

(19) 日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開 2001-169278

(P 2001-169278 A)

(43) 公開日 平成13年6月22日 (2001. 6. 22)

(51) Int. Cl.<sup>7</sup>

識別記号

F I

テマコード\* (参考)

H 0 4 N 7/24  
5/91  
5/92  
7/08  
7/081H 0 4 N 7/13  
5/91  
5/92  
7/08  
7/137Z 5C053  
N 5C059  
H 5C063  
Z  
Z

審査請求 未請求 請求項の数 14 O L

(全 57 頁)

最終頁に続く

(21) 出願番号 特願平11-346330

(22) 出願日 平成11年12月6日 (1999. 12. 6)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 北村 卓也

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(72) 発明者 田原 勝己

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(74) 代理人 100082131

弁理士 稲本 義雄

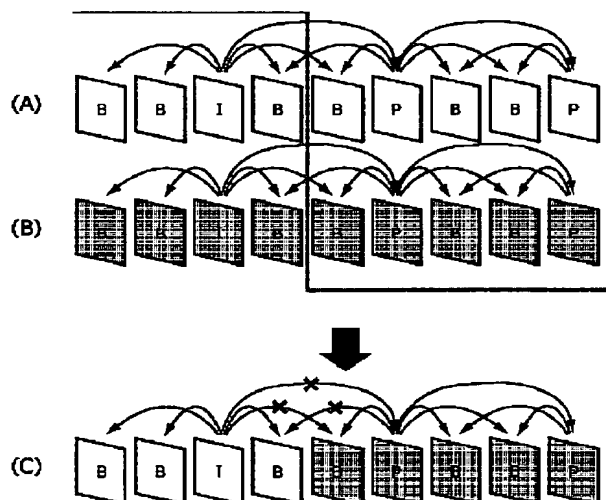
最終頁に続く

(54) 【発明の名称】 ストリーム生成装置および方法、ストリーム伝送装置および方法、符号化装置および方法、並びに記録媒体

(57) 【要約】

【課題】 画質劣化の発生しないトランスコーディングシステムを実現する。

【解決手段】 ヒストリ情報分離装置 115 は、過去の符号化処理における符号化履歴を検出する。比較装置 114 は、画像の不連続を検出する。符号化装置 116 は、過去の符号化処理における符号化履歴および画像の不連続を基に、符号化ストリームを生成する。



## 【特許請求の範囲】

【請求項 1】 入力された第 1 のストリームから第 2 のストリームを生成し、出力するストリーム生成装置において、

前記第 1 のストリームの過去の符号化処理における符号化履歴を検出する第 1 の検出手段と、

前記第 1 のストリームの画像の不連続を検出する第 2 の検出手段と、

前記第 1 の検出手段の検出結果および前記第 2 の検出手段の検出結果を利用して、前記第 1 のストリームを基に前記第 2 のストリームを生成する生成手段とを含むことを特徴とするストリーム生成装置。

【請求項 2】 前記生成手段は、MPEG方式で符号化して、前記第 2 のストリームを生成することを特徴とする請求項 1 に記載のストリーム生成装置。

【請求項 3】 前記第 2 のストリームに対して、それまでの符号化処理における符号化履歴を記述する第 1 の記述手段を更に含むことを特徴とする請求項 1 に記載のストリーム生成装置。

【請求項 4】 前記第 1 の記述手段は、前記符号化履歴の情報を、前記MPEG方式の前記第 2 のストリームのuser\_dataとして記述することを特徴とする請求項 3 に記載のストリーム生成装置。

【請求項 5】 前記第 2 のストリームに対して、画像の連続を表す連続情報を記述する第 2 の記述手段を更に含むことを特徴とする請求項 1 に記載のストリーム生成装置。

【請求項 6】 前記第 2 の記述手段は、前記画像の連続を表す前記連続情報を、前記MPEG方式の前記第 2 のストリームのuser\_dataとして記述することを特徴とする請求項 5 に記載のストリーム生成装置。

【請求項 7】 前記第 2 の記述手段は、アクセスユニット毎に加算または減算により連続情報を生成して記述することを特徴とする請求項 5 に記載のストリーム生成装置。

【請求項 8】 前記第 2 の記述手段は、復号された画像信号の輝度信号または色差信号の所定のビットのランキング部分に前記連続情報を記述することを特徴とする請求項 5 に記載のストリーム生成装置。

【請求項 9】 前記第 2 の記述手段は、復号された画像信号のランキング部分に前記連続情報を記述することを特徴とする請求項 5 に記載のストリーム生成装置。

【請求項 10】 前記連続情報は、パケット化されていることを特徴とする請求項 5 に記載のストリーム生成装置。

【請求項 11】 前記第 2 の記述手段は、復号された画像信号の輝度信号または色差信号の所定のビットのランキング部分に多重化されている前記符号化履歴の情報を、前記連続情報を多重化することを特徴とする請求項 5 に記載のストリーム生成装置。

【請求項 12】 前記第 2 の記述手段は、復号された画像信号のランキング部分に多重化されている前記符号化履歴の情報を、前記連続情報を多重化することを特徴とする請求項 5 に記載のストリーム生成装置。

【請求項 13】 入力された第 1 のストリームから第 2 のストリームを生成し、出力するストリーム生成装置のストリーム生成方法において、

前記第 1 のストリームの過去の符号化処理における符号化履歴を検出する第 1 の検出ステップと、

前記第 1 のストリームの画像の不連続を検出する第 2 の検出ステップと、

前記第 1 の検出ステップの処理での検出結果および前記第 2 の検出ステップの処理での検出結果を利用して、前記第 1 のストリームを基に前記第 2 のストリームを生成する生成ステップとを含むことを特徴とするストリーム生成方法。

【請求項 14】 入力された第 1 のストリームから第 2 のストリームを生成し、出力するストリーム生成装置を制御するプログラムにおいて、

前記第 1 のストリームの過去の符号化処理における符号化履歴を検出する第 1 の検出ステップと、

前記第 1 のストリームの画像の不連続を検出する第 2 の検出ステップと、

前記第 1 の検出ステップの処理での検出結果および前記第 2 の検出ステップの処理での検出結果を利用して、前記第 1 のストリームを基に前記第 2 のストリームを生成する生成ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、ストリーム生成装置および方法、ストリーム伝送装置および方法、符号化装置および方法、並びに記録媒体に関し、特に、MPEG規格に基づいて符号化された符号化ビットストリームのGOP (Group of Pictures) の構造を変更したり、符号化ビットストリームのビットレートを変更するためのトランスコーディング装置に用いて好適なストリーム生成装置および方法、ストリーム伝送装置および方法、符号化装置および方法、並びに記録媒体に関する。

## 【0002】

【従来の技術】近年、テレビジョンプログラムを制作及び放送する放送局においては、ビデオデータを圧縮/符号化処理するために、MPEG (Moving Picture Experts Group) 技術が一般的に使われるようになってきた。特に、ビデオデータをテープなどのランダムアクセス可能な記録媒体素材に記録する場合、及びビデオデータをケーブルや衛星を介して伝送する場合には、このMPEG技術がデファクトスタンダードになりつつある。

【0003】放送局において制作されたビデオプログラ

ムが各家庭に伝送されるまでの放送局における処理の一例を簡単に説明する。まず、ビデオカメラとVTR (Video Tape Recorder) が一体となったカムコーダに設けられたエンコーダによって、ソースビデオデータをエンコード処理して磁気テープ上に記録する。この際、カムコーダのエンコーダは、VTRのテープの記録フォーマットに適するように、ソースビデオデータを符号化する。たとえば、この磁気テープ上に記録されるMPEGビットストリームのGOP構造は、2 フレームから1 GOPが構成される構造 (たとえば、I, B, I, B, I, B, ……) とされる。また磁気テープ上に記録されているMPEGビットストリームのビットレートは、1 8 Mbpsである。

【0 0 0 4】次に、メイン放送局において、この磁気テープ上に記録されたビデオビットストリームを編集する編集処理を行う。そのために、磁気テープ上に記録されたビデオビットストリームのGOP構造を、編集処理に適したGOP構造に変換する。編集処理に適したGOP構造とは、1 GOPが1 フレームから構成され、すべてのピクチャがIピクチャであるGOP構造である。なぜなら、フレーム単位で編集を行うためには、他のピクチャと相関のないIピクチャがもっとも適しているからである。実際のオペレーションとしては、磁気テープ上に記録されたビデオストリームを一旦デコードしてベースバンドのビデオデータに戻す。そして、そのベースバンドのビデオ信号を、すべてのピクチャがIピクチャとなるように再エンコードする。このようにデコード処理及び再エンコード処理を行うことによって、編集処理に適したGOP構造を有したビットストリームを生成することができる。

【0 0 0 5】次に、上述した編集処理によって生成された編集ビデオプログラムを、メイン局から地方局に伝送するために、編集ビデオプログラムのビットストリームを、伝送処理に適したGOP構造及びビットレートに変換する。放送局間の伝送に適したGOP構造とは、たとえば、1 GOPが1 5 フレームから構成されているGOP構造 (たとえば、I, B, B, P, B, B, P…) である。また、放送局間の伝送に適したビットレートは、一般的に放送局間においては、光ファイバなどの高伝送容量を有した専用線が設けられているので、5 0 Mbps以上のハイビットレートであることが望ましい。具体的には、編集処理されたビデオプログラムのビットストリームを一旦デコードしてベースバンドのビデオデータに戻す。そして、そのベースバンドのビデオデータを上述した放送局間の伝送に適したGOP構造及びビットレートを有するように再エンコードする。

【0 0 0 6】地方局においては、メイン局から伝送されてきたビデオプログラムの中に、地方特有の商業的挿入のために編集処理が行われる。つまり、上述した編集処理と同じように、メイン局から伝送されてきたビデオストリームを一旦デコードしてベースバンドのビデオデータに戻す。そして、そのベースバンドのビデオ

オ信号を、すべてのピクチャがIピクチャとなるように再エンコードすることによって、編集処理に適したGOP構造を有したビットストリームを生成することができる。

【0 0 0 7】続いて、この地方局において編集処理が行われたビデオプログラムを各家庭に、ケーブルや衛星を介して伝送するために、この伝送処理に適したGOP構造及びビットレートに変換する。たとえば、各家庭に伝送するための伝送処理に適したGOP構造とは、1 GOPが1 5 フレームから構成されるGOP構造 (たとえば、I, B, B, P, B, B, P…) であって、各家庭に伝送するための伝送処理に適したビットレートは、5 Mbps程度の低ビットレートである。具体的には、編集処理されたビデオプログラムのビットストリームを一旦デコードしてベースバンドのビデオデータに戻す。そして、そのベースバンドのビデオデータを上述した伝送処理に適したGOP構造及びビットレートを有するように再エンコードする。

【0 0 0 8】このように、放送局から各家庭にビデオプログラムが伝送される間に、複数回の復号処理、符号化処理、および編集処理が繰り返されている。実際には、放送局における処理は上述した信号処理以外にもさまざまな信号処理が必要であり、そのたびに復号処理及び符号化処理を繰り返される。

【0 0 0 9】MPEG規格に基づく符号化処理及び復号処理は、1 0 0 %可逆の処理ではないことは良く知られている。つまり、エンコードされる前のベースバンドのビデオデータと、デコードされた後のビデオデータは1 0 0 %同じでは無く、この符号化処理及び復号処理によって画質が劣化している。つまり、上述したように、デコード処理及びエンコード処理を繰り返すと、その処理の度に、画質が劣化してしまうという問題があった。別の言葉で表現すると、デコード/エンコード処理を繰り返す毎に、画質の劣化が蓄積されてしまう。

【0 0 1 0】そこで、デコード/エンコード処理に伴う画質の劣化を防止するために、以前の符号化の処理で用いられた符号化パラメータを画像と関連付けて伝送して、以前の符号化の処理で用いられた符号化パラメータを利用して符号化するシステムが利用されている。

【0 0 1 1】

【発明が解決しようとする課題】しかしながら、画像の削除または画像の挿入などの編集処理が行われた画像を、以前の符号化の処理で用いられた符号化パラメータを利用して符号化しようとする場合、大きな画質劣化を招くときがある。

【0 0 1 2】図1は、時間の経過とともに図1中の左側から右側に向かって表示されるピクチャの例を示す。図1中のBは、以前の符号化の処理においてBピクチャとして符号化されたピクチャを示し、図1中のIは、以前の符号化の処理においてIピクチャとして符号化された

ピクチャを示し、図1中のPは、以前の符号化の処理においてPピクチャとして符号化されたピクチャを示す。

【0013】例えば、図1(A)に示す図1(A)中の左側の4つのピクチャに、図1(B)に示す図1(B)中の右側の5つのピクチャがつなぎ合わされて、図1

(C)に示す新たな画像として生成されるように編集されたとき、編集のつなぎ目の最後の画像(編集前に図1(A)に含まれていた、図1(A)中左側から4枚目の画像に対応する画像)が、以前の符号化においてBピクチャである場合、その以前の符号化で用いられた符号化パラメータを利用して符号化しようとするとき、編集される前とは異なるPピクチャを参照して符号化してしまい、大きく画質が劣化してしまう。

【0014】同様に、編集のつなぎ目の先頭の画像(編集前に図1(B)に含まれていた、図1(B)中左側から5枚目の画像に対応する画像)が、以前の符号化においてBピクチャである場合、その以前の符号化で用いられた符号化パラメータを利用して符号化しようとするとき、編集される前とは異なるIピクチャを参照して符号化してしまい、大きく画質が劣化してしまう。

【0015】また、このような編集が行われた場合、VBV(Video Buffering Verifier) Bufferを基にした、ビットストリームのレートコントロールに矛盾が生じるので、オーバーフローまたはアンダーフローが生じるときがある。

【0016】例えば、図2(A)が、以前の符号化の処理において、図1(A)に示す画像を符号化する場合の、VBV Bufferに格納されるデータの量を示し、図2

(B)が、以前の符号化の処理において、図1(B)に示す画像を符号化する場合の、VBV Bufferに格納されるデータの量を示すとき、図1(C)に示す画像を、以前の符号化パラメータをそのまま利用して符号化すると、図2(C)に示すように、VBV Bufferに格納されるデータは、オーバーフローしてしまう。

【0017】更に、フレームシンクロナイザにより、フレーム周期のずれを吸収するため、画像のフレームを間引いたり、または同一のフレームを二回表示させた場合にも、画像の時間的な連続性を壊すことになり、上述の編集処理の場合と同様の問題が生じる。

【0018】MPEG2のビットストリームにおいて、入力画像毎にカウントアップする、10bitのカウンタであるtemporal\_referenceがpicture\_header()層に挿入されているので、このtemporal\_referenceを利用して、このような画像の不連続を検出することができる。

【0019】しかし、このtemporal\_referenceは、group\_of\_pictures\_header()の後に、リセットしなければならない。group\_of\_pictures\_header()は、周期について特に規定はないが、通常、GOPの周期で挿入されている。

【0020】従って、1GOPが15フレームから構成さ

れるGOP構造を有する場合、temporal\_referenceの周期は15となり、この画像に上述のような編集処理がなされても、temporal\_referenceは連続してしまう可能性が高いので、画像の不連続を検出できない可能性が高い。

1GOPが1フレームから構成されるGOP構造を有する場合、temporal\_referenceは、常に、0となるので、画像の不連続は検出できない。

【0021】本発明は、このような状況に鑑みてなされたものであり、MPEG規格に基づいて符号化された符号化ビットストリームのGOP(Group of Pictures)の構造を変更するために復号処理、符号化処理、および編集処理を繰り返したとしても画質劣化の発生しないようにするものである。

【0022】

【課題を解決するための手段】請求項1に記載のストリーム生成装置は、第1のストリームの過去の符号化処理における符号化履歴を検出する第1の検出手段と、第1のストリームの画像の不連続を検出する第2の検出手段と、第1の検出手段の検出結果および第2の検出手段の検出結果を利用して、第1のストリームを基に第2のストリームを生成する生成手段とを含むことを特徴とする。

【0023】生成手段は、MPEG方式で符号化して、第2のストリームを生成するようにすることができる。

【0024】ストリーム生成装置は、第2のストリームに対して、それまでの符号化処理における符号化履歴を記述する第1の記述手段を更に設けることができる。

【0025】第1の記述手段は、符号化履歴の情報を、MPEG方式の第2のストリームのuser\_dataとして記述するようにすることができる。

【0026】ストリーム生成装置は、第2のストリームに対して、画像の連続を表す連続情報を記述する第2の記述手段を更に設けることができる。

【0027】第2の記述手段は、画像の連続を表す連続情報を、MPEG方式の第2のストリームのuser\_dataとして記述するようにすることができる。

【0028】第2の記述手段は、アクセスユニット毎に加算または減算により連続情報を生成して記述するようにすることができる。

【0029】第2の記述手段は、復号された画像信号の輝度信号または色差信号の所定のビットのブランキング部分に連続情報を記述するようにすることができる。

【0030】第2の記述手段は、復号された画像信号のブランキング部分に連続情報を記述するようにすることができる。

【0031】ストリーム生成装置は、連続情報をパケット化するようにすることができる。

【0032】第2の記述手段は、復号された画像信号の輝度信号または色差信号の所定のビットのブランキング部分に多重化されている符号化履歴の情報に、連続情報

を多重化するようにすることができる。

【0033】第2の記述手段は、復号された画像信号のブランキング部分に多重化されている符号化履歴の情報に、連続情報を多重化するようにすることができる。

【0034】請求項13に記載のストリーム生成方法は、第1のストリームの過去の符号化処理における符号化履歴を検出する第1の検出ステップと、第1のストリームの画像の不連続を検出する第2の検出ステップと、第1の検出ステップの処理での検出結果および第2の検出ステップの処理での検出結果を利用して、第1のストリームを基に第2のストリームを生成する生成ステップとを含むことを特徴とする。

【0035】請求項14に記載の記録媒体のプログラムは、第1のストリームの過去の符号化処理における符号化履歴を検出する第1の検出ステップと、第1のストリームの画像の不連続を検出する第2の検出ステップと、第1の検出ステップの処理での検出結果および第2の検出ステップの処理での検出結果を利用して、第1のストリームを基に第2のストリームを生成する生成ステップとを含むことを特徴とする。

【0036】請求項1に記載のストリーム生成装置、請求項13に記載のストリーム生成方法および請求項14に記載の記録媒体においては、第1のストリームの過去の符号化処理における符号化履歴が検出され、第1のストリームの画像の不連続が検出され、符号化履歴の検出結果および画像の不連続の検出結果を利用して、第1のストリームを基に第2のストリームが生成される。

#### 【0037】

【発明の実施の形態】以下に、本発明を適用したトランスコーディングシステムについて説明する。なお、本明細書においてシステムの用語は、複数の装置、手段などにより構成される全体的な装置を意味するものである。

【0038】図3及び図4は、本発明を適用したトランスコーディングシステム1の構成を示しており、図4は、図3のさらに詳細な構成を示している。このトランスコーディングシステム1は、ビデオ復号システム11に入力された符号化ビデオビットストリーム(encoded video bit stream)のGOP構造及びビットレートを、オペレータの所望するGOP構造及びビットレートに変換する。

【0039】トランスコーディングシステム1は、ビデオ復号システム11、ビデオ符号化システム12、VTR (Video Tape Recorder) 13、スイッチ14、およびスイッチ15から構成されている。

【0040】ビデオ復号システム11は、入力された符号化ビデオビットストリームを基にベースバンドデジタルビデオ信号を生成する。ビデオ符号化システム12は、ベースバンドビデオ信号を基にオペレータの所望するGOP構造及びビットレートを有する符号化ビデオビットストリームを出力する。

【0041】VTR13は、スイッチ14を介して供給されたベースバンドデジタルビデオ信号を記録すると共に、記録されているベースバンドデジタルビデオ信号をスイッチ15を介してビデオ符号化システム12に供給する。

【0042】スイッチ14は、ビデオ復号システム11から出力されたベースバンドデジタルビデオ信号の供給先をVTR13またはスイッチ15に切り換える。スイッチ15は、ビデオ復号システム11から出力されるベースバンドデジタルビデオ信号またはVTR13から出力されるベースバンドデジタルビデオ信号のいずれかを選択して、選択されたベースバンドデジタルビデオ信号をビデオ符号化システム12に出力する。

【0043】このトランスコーディングシステム1の機能を説明するために、図4には図示されていないが、このトランスコーディングシステム1の前段に、このトランスコーディングシステム1とほぼ同様の機能を有した3つのトランスコーディングシステムが接続されているものとする。つまり、ビットストリームのGOP構造及びビットレートをさまざまに変更するために、第1のトランスコーディングシステム、第2のトランスコーディングシステム、および第3のトランスコーディングシステムが順に直列に接続され、その第3のトランスコーディングシステムの後ろに、この図4に示された第4のトランスコーディングシステムが接続されているものとする。

【0044】本発明の以下の説明において、この第1のトランスコーディングシステムにおいて行われた符号化処理を第1世代の符号化処理と定義し、第1のトランスコーディングシステムの後ろに接続された第2のトランスコーディングシステムにおいて行われた符号化処理を第2世代の符号化処理と定義し、第2のトランスコーディングシステムの後ろに接続された第3のトランスコーディングシステムにおいて行われた符号化処理を第3世代の符号化処理と定義し、第3のトランスコーディングシステムの後ろに接続された第4のトランスコーディングシステム(図4に示されたトランスコーディングシステム1)において行われる符号化処理を第4世代の符号化処理または現在の符号化処理と定義することにする。

【0045】また、第1世代の符号化処理において生成された符号化パラメータを第1世代の符号化パラメータと呼び、第2世代の符号化処理において生成された符号化パラメータを第2世代の符号化パラメータと呼び、第3世代の符号化処理において生成された符号化パラメータを第3世代の符号化パラメータと呼び、第4世代の符号化処理において生成された符号化パラメータを第4世代の符号化パラメータまたは現在の符号化パラメータと呼ぶことにする。

【0046】まず、この図4に示されたトランスコーディングシステム1に供給される符号化ビデオストリーム

ST (3rd) について説明する。ST (3rd) は、このトランスコーディングシステム 1 の前段に設けられている第 3 のトランスコーディングシステムにおける第 3 世代の符号化処理において生成された第 3 世代の符号化ストリームであることを表わしている。この第 3 世代の符号化処理において生成された符号化ビデオストリーム ST (3rd) には、第 3 の符号化処理において生成された第 3 世代の符号化パラメータが、この符号化符号化ビデオストリーム ST (3rd) のシーケンス層、GOP 層、ピクチャ層、スライス層、及びマクロブロック層に、sequence\_header() 関数、sequence\_extension() 関数、group\_of\_picture\_header() 関数、picture\_header() 関数、picture\_coding\_extension() 関数、picture\_data() 関数、slice() 関数、及びmacroblock() 関数として記述されている。このように第 3 の符号化処理によって生成された第 3 の符号化ストリームに、第 3 の符号化処理において使用した第 3 の符号化パラメータを記述することは MPEG 2 規格において定義されていることであって、何ら新規性は無い。

【0047】本発明のトランスコーディングシステム 1 におけるユニークな点は、この第 3 の符号化ストリーム ST (3rd) 中に、第 3 の符号化パラメータを記述するだけでなく、第 1 世代及び第 2 世代の符号化処理において生成された第 1 世代及び第 2 世代の符号化パラメータも記述されているという点、およびアクセスユニットであるフレームまたはフィールド毎にカウントアップされる、十分に周期の長いカウンタ値が画像に関連づけられている点である。

【0048】具体的には、この第 1 世代及び第 2 世代の符号化パラメータは、第 3 世代の符号化ビデオストリーム ST (3rd) のピクチャ層のユーザデータエリアに、ヒストリストリーム history\_stream() として記述されている。本発明においては、第 3 世代の符号化ビデオストリーム ST (3rd) のピクチャ層のユーザデータエリアに記述されているヒストリストリームを、「ヒストリ情報」、または「履歴情報」と呼び、このヒストリストリームとして記述されている符号化パラメータを「ヒストリパラメータ」、または「履歴パラメータ」と呼んでいる。

【0049】また別の呼び方として、第 3 世代の符号化ストリーム ST (3rd) に記述されている第 3 世代の符号化パラメータを「現在の符号化パラメータ」と呼んだ場合には、第 3 世代の符号化処理からみて第 1 世代及び第 2 世代の符号化処理は、過去に行なわれた符号化処理であるので、第 3 世代の符号化ストリーム ST (3rd) のピクチャ層のユーザデータエリアに記述されているヒストリストリームとして記述されている符号化パラメータを「過去の符号化パラメータ」とも呼んでいる。

【0050】このように、この第 3 の符号化ストリーム ST (3rd) 中に、第 3 の符号化パラメータを記述する

だけでなく、第 1 世代及び第 2 世代の符号化処理において生成された第 1 世代及び第 2 世代の符号化パラメータを記述する理由は、トランスコーディング処理によって符号化ストリームの GOP 構造やビットレートの変更を繰り返したとしても、画質劣化を防止することができるからである。

【0051】例えば、あるピクチャを第 1 世代の符号化処理において P ピクチャとして符号化し、第 1 世代の符号化ストリームの GOP 構造を変更するために、第 2 世代の符号化処理においてそのピクチャを B ピクチャとして符号化し、第 2 世代の符号化ストリームの GOP 構造をさらに変更するために、第 3 世代の符号化処理において、再度そのピクチャを P ピクチャとして符号化することが考えられる。MPEG 規格に基づく符号化処理及び復号処理は 100% 可逆の処理ではないので、符号化及び復号処理を繰り返す毎に画質が劣化していくことは知られている。

【0052】このような場合に、第 3 の世代の符号化処理において、量子化スケール、動きベクトル、予測モードなどの符号化パラメータをもう一度計算するのではなくて、第 1 世代の符号化処理において生成された量子化スケール、動きベクトル、予測モードなどの符号化パラメータを再利用する。第 3 世代の符号化処理によって新しく生成された量子化スケール、動きベクトル、予測モードなどの符号化パラメータよりも、第 1 世代の符号化処理によって新しく生成された量子化スケール、動きベクトル、予測モードなどの符号化パラメータの方が、明らかに精度が良いので、この第 1 世代のパラメータを再利用することによって、符号化及び復号処理を繰り返したとしても画質劣化を少なくすることができる。

【0053】また、ビデオ復号システム 11 が出力するベースバンドデジタルビデオ信号に含まれる画像には、アクセスユニットであるそのフレームまたはフィールド毎にカウントアップされるカウンタ値が関連付けられている。カウンタ値は、例えば、周期が 65,536 であり、最大値が設定された後、0 になり、また、0 からカウントアップされる。

【0054】ビデオ符号化システム 12 は、ベースバンドデジタルビデオ信号に含まれる画像に対応するカウンタ値を基に、入力されたベースバンドデジタルビデオ信号に含まれる画像の不連続点（例えば、画像をつなぎ合わせた点、画像を挿入した点、画像を間引いた点など）を検出することができる。

【0055】例えば、あるピクチャを第 1 世代の符号化処理において P ピクチャとして符号化し、第 1 世代の符号化ストリームの GOP 構造を変更するために、第 2 世代の符号化処理においてそのピクチャを B ピクチャとして符号化し、第 2 世代の符号化ストリームの GOP 構造をさらに変更するために、第 3 世代の符号化処理において、第 1 世代の符号化処理における符号化パラメータを基に

再度そのピクチャをPピクチャとして符号化する場合、そのピクチャの前のピクチャが削除されるように（そのピクチャが参照するIピクチャまたはPピクチャが削除されるように）画像が編集されていないとき、ビデオ符号化システム12は、第1世代の符号化処理において生成した符号化パラメータを利用してそのピクチャをPピクチャとして符号化し、そのピクチャの前のピクチャが削除されるように（そのピクチャが参照するIピクチャまたはPピクチャが削除されるように）画像が編集されているとき、符号化パラメータを生成してそのピクチャをPピクチャとして符号化する。

【0056】このように、ビデオ符号化システム12は、アクセスユニットであるフレームまたはフィールド毎にカウントアップされるカウンタ値を基に画像の不連続点を検出して、第1世代の符号化処理において生成された量子化スケール、動きベクトル、予測モードなどの符号化パラメータを利用して、符号化するので、画質劣化を防止することができる。

【0057】上述した本発明にかかる処理を説明するために、図4に示された第4世代のトランスコーディングシステム1の処理を例に挙げてより詳しく説明する。

【0058】カウンタ101は、復号装置102から供給されるフレームまたはフィールドに同期したFrame/Field同期信号を基に、カウントアップ（1を加算）される16ビットのカウンタである。カウンタ101は、0乃至65,535のいずれかのカウンタ値をカウンタ値多重化装置105に出力する。

【0059】カウンタ101は、65,535のカウンタ値を有する場合、復号装置102からフレームまたはフィールドに同期したFrame/Field同期信号が供給されたとき、カウンタ値を0とし、その後も、復号装置102から供給されるFrame/Field同期信号を基に、カウントアップを継続する。

【0060】なお、カウンタ101は、復号装置102から供給されるフレームまたはフィールドに同期したFrame/Field同期信号を基に、カウンタダウン（1を減算）するようにしてもよい。

【0061】復号装置102は、第3世代の符号化ビットストリームST（3rd）に含まれている符号化ビデオを第3世代の符号化パラメータを使用して復号し、復号されたベースバンドのデジタルビデオデータを生成するための装置である。さらに、復号装置102は、第3世代の符号化ビットストリームST（3rd）のピクチャ層のユーザデータエリアにヒストリストリームとして記述されている第1世代及び第2世代の符号化パラメータをデコードするための装置でもある。

【0062】具体的には、図5に示されているように、復号装置102のデコーダ251は、供給されたビットストリームをバッファリングするための受信バッファ261、符号化ビットストリームを可変長復号するための

可変長復号回路262、可変長復号されたデータを可変長復号回路262から供給された量子化スケールに従って逆量子化する逆量子化回路263、逆量子化されたDCT（離散コサイン変換）係数を逆離散コサイン変換するIDCT回路264、及び動き補償処理を行うための演算器265、動き補償回路266及びフレームメモリ267を備えている。

【0063】伝送路（または所定の記録媒体）を介して伝送された符号化された画像データは、図示せぬ受信回路で受信されたり、再生装置で再生され、受信バッファ261に一時記憶された後、可変長復号回路262に供給される。可変長復号回路262は、受信バッファ261より供給されたデータを可変長復号し、動きベクトル、予測モード、予測フラグ、およびDCTフラグを動き補償回路266に出力し、量子化スケールを逆量子化回路263に出力するとともに、復号された画像データを逆量子化回路263に出力する。

【0064】逆量子化回路263は、可変長復号回路262より供給された画像データを、同じく可変長復号回路262より供給された量子化スケールに従って逆量子化し、IDCT回路264に出力する。逆量子化回路263より出力されたデータ（DCT係数）は、IDCT回路264により、逆離散コサイン変換処理が施され、演算器265に供給される。

【0065】IDCT回路264より演算器265に供給された画像データが、Iピクチャのデータである場合、そのデータは演算器265より出力され、演算器265に後に入力される画像データ（PまたはBピクチャのデータ）の予測画像データ生成のために、フレームメモリ267の前方予測画像部267aに供給されて記憶される。また、このデータは、ヒストリ情報多重化装置103（図3）に出力される。

【0066】IDCT回路264より供給された画像データが、その1フレーム前の画像データを予測画像データとするPピクチャのデータであり、前方予測モードのデータである場合、フレームメモリ267の前方予測画像部267aに記憶されている、1フレーム前の画像データ（Iピクチャのデータ）が読み出され、動き補償回路266で可変長復号回路262より出力された動きベクトルに対応する動き補償が施される。そして、演算器265において、IDCT回路264より供給された画像データ（差分のデータ）と加算され、出力される。この加算されたデータ、すなわち、復号されたPピクチャのデータは、演算器265に後に入力される画像データ（BピクチャまたはPピクチャのデータ）の予測画像データ生成のために、フレームメモリ267の後方予測画像部267bに供給されて記憶される。

【0067】Pピクチャのデータであっても、画像内予測モードのデータは、Iピクチャのデータと同様に、演算器265において処理は行われず、そのまま後方予測

画像部 2 6 7 b に記憶される。

【0 0 6 8】この P ピクチャは、次の B ピクチャの次に表示されるべき画像であるため、この時点では、まだヒストリ情報多重化装置 1 0 3 へ出力されない（B ピクチャの後に入力された P ピクチャが、B ピクチャより先に処理され、伝送される）。

【0 0 6 9】IDCT 回路 2 6 4 より供給された画像データが、B ピクチャのデータである場合、可変長復号回路 2 6 2 より供給された予測モードに対応して、フレームメモリ 2 6 7 の前方予測画像部 2 6 7 a に記憶されている I ピクチャの画像データ（前方予測モードの場合）、後方予測画像部 2 6 7 b に記憶されている P ピクチャの画像データ（後方予測モードの場合）、または、その両方の画像データ（両方向予測モードの場合）が読み出され、動き補償回路 2 6 6 において、可変長復号回路 2 6 2 より出力された動きベクトルに対応する動き補償が施されて、予測画像が生成される。但し、動き補償を必要としない場合（画像内予測モードの場合）、予測画像は生成されない。

【0 0 7 0】このようにして、動き補償回路 2 6 6 で動き補償が施されたデータは、演算器 2 6 5 において、IDCT 回路 2 6 4 の出力と加算される。この加算出力は、ヒストリ情報多重化装置 1 0 3 に出力される。

【0 0 7 1】ただし、この加算出力は B ピクチャのデータであり、他の画像の予測画像生成のために利用されることがないため、フレームメモリ 2 6 7 には記憶されない。

【0 0 7 2】B ピクチャの画像が出力された後、後方予測画像部 2 6 7 b に記憶されている P ピクチャの画像データが読み出され、動き補償回路 2 6 6 を介して演算器 2 6 5 に供給される。但し、このとき、動き補償は行われない。

【0 0 7 3】なお、このデコード 2 5 1 において、奇数フィールドと偶数フィールドのラインの信号が分離された構成を元の構成に必要なに応じて戻す処理は、動き補償回路 2 6 6 により実行される。

【0 0 7 4】また、上述した説明においては、輝度信号の処理について説明したが、色差信号の処理も同様に行われる。ただし、この場合の動きベクトルは、輝度信号用の動きベクトルを、垂直方向および水平方向に 1 / 2 にしたものが用いられる。

【0 0 7 5】可変長復号回路 2 6 2 は、第 3 世代の符号化ビットストリーム S T (3rd) を復号処理するために、この第 3 世代の符号化ビットストリーム S T (3rd) のピクチャ層、スライス層及びマクロブロック層に記述されている第 3 世代の符号化パラメータを抽出する。たとえば、この可変長復号回路 2 6 2 において抽出される第 3 世代の符号化パラメータは、ピクチャタイプを示す picture\_coding\_type、量子化スケールステップサイズを示す quantiser\_scale\_code、予測モードを示す

macroblock\_type、動きベクトルを示す motion\_vector、Frame 予測モードか Field 予測モードかを示す frame/field\_motion\_type、及び Frame D C T モードか Field D C T モードかを示す dct\_type 等である。この可変長復号回路 2 6 2 において抽出された quantiser\_scale\_code は、逆量子化回路 2 6 3 に供給され、picture\_coding\_type、quantiser\_scale\_code、macroblock\_type、motion\_vector、frame/field\_motion\_type、dct\_type 等のパラメータは、動き補償回路 2 6 6 に供給される。

10 【0 0 7 6】可変長復号回路 2 6 2 は、第 3 世代の符号化ビットストリーム S T (3rd) を復号処理するために必要なこれらの符号化パラメータだけではなく、後段の第 5 世代のトランスコーディングシステムに第 3 世代のヒストリ情報として伝送されるべき符号化パラメータを、第 3 世代の符号化ビットストリーム S T (3rd) のシーケンス層、GOP 層、ピクチャ層、スライス層、及びマクロブロック層から抽出する。もちろん、第 3 世代の復号処理に使用された picture\_coding\_type、quantiser\_scale\_code、macroblock\_type、motion\_vector、frame/field\_motion\_type、dct\_type 等の第 3 世代の符号化パラメータは、この第 3 世代のヒストリ情報に含まれている。ヒストリ情報としてどのような符号化パラメータを抽出するかについては、伝送容量などに応じてオペレータやホストコンピュータ側からあらかじめ設定されている。

20 【0 0 7 7】また、可変長復号回路 2 6 2 は、アクセスユニットであるフレームまたはフィールドの切り換えに対応する Frame/Field 同期信号をカウンタ 1 0 1 に供給する。

30 【0 0 7 8】さらに、可変長復号回路 2 6 2 は、第 3 世代の符号化ビットストリーム S T (3rd) のピクチャ層のユーザデータエリアに記述されているユーザデータを抽出し、そのユーザデータをヒストリデコーディング装置 1 0 4 に供給する。

40 【0 0 7 9】このヒストリデコーディング装置 1 0 4 は、第 3 世代の符号化ビットストリーム S T (3rd) のピクチャ層に記述されていたユーザデータから、ヒストリ情報として記述されている第 1 世代の符号化パラメータ及び第 2 世代の符号化パラメータ（直前の世代よりさらに前の世代の符号化パラメータ）を抽出するための回路である。具体的には、ヒストリデコーディング装置 1 0 4 は、受け取ったユーザデータのシンタックスを解析することによって、ユーザデータの中に記述されている固有の History\_Data\_Id を検出し、これによって、converted\_history\_stream() を抽出することができる。さらに、ヒストリデコーディング装置 1 0 4 は、converted\_history\_stream() 中にある所定間隔に挿入されている 1 ビットのマーカービット (marker\_bit) を取りさることによって、history\_stream() を得、そして、その history\_stream() のシンタックスを解析することによって、hi



story\_stream()中に記述されている第1世代及び第2世代の符号化パラメータを得ることができる。このヒストリデコーディング装置104の詳しい動作については、後述する。

【0080】ヒストリ情報多重化装置103は、第1世代、第2世代及び第3世代の符号化パラメータを、第4世代の符号化処理を行う符号化装置116に供給するために、復号装置102においてデコードされたベースバンドのビデオデータに、これらの第1世代、第2世代及び第3世代の符号化パラメータを多重化するための回路である。具体的には、ヒストリ情報多重化装置103は、復号装置102の演算器265から出力されたベースバンドのビデオデータ、復号装置102の可変長復号装置112から出力された第3世代の符号化パラメータ、並びに、ヒストリデコーディング装置104から出力された第1世代の符号化パラメータと第2世代の符号化パラメータとを受け取り、このベースバンドのビデオデータに、これらの第1世代、第2世代及び第3世代の符号化パラメータを多重化する。第1世代、第2世代及び第3世代の符号化パラメータが多重化されたベースバンドのビデオデータは、カウンタ値多重化装置105に供給される。

【0081】カウンタ値多重化装置105は、ヒストリ情報多重化装置103から供給された、第1世代、第2世代及び第3世代の符号化パラメータが多重化されたベースバンドのビデオデータに、更に、カウンタ101から供給されたカウンタ値を多重化する。

【0082】次に、これらの第1世代、第2世代及び第3世代の符号化パラメータのベースバンドビデオデータへの多重化の方法について、図6及び図7を参照して説明する。図6は、MPEG規格において定義されている、16ピクセル×16ピクセルからなる1つのマクロブロックを示している。この16ピクセル×16ピクセルのマクロブロックは、輝度信号に関しては4つの8ピクセル×8ピクセルからなるサブブロック(Y[0], [1], [2]及びY[3])と、色差信号に関しては4つの8ピクセル×8ピクセルからなるサブブロック(Cr[0], r[1], b[0], 及びCb[1])から構成されている。

【0083】図7は、ビデオデータのあるフォーマットを表している。このフォーマットは、ITU勧告-RDT601において定義されているフォーマットであって、放送業界において使用されている所謂「D1フォーマット」を表している。このD1フォーマットは、10ビットのビデオデータを伝送するためのフォーマットとして規格化されたので、ビデオデータの1ピクセルを10ビットで表現できるようになっている。

【0084】MPEG規格によってデコードされたベースバンドのビデオデータは8ビットであるので、本発明のトランスコーディングシステムにおいては、図7に示したように、D1フォーマットの10ビットのうち上位8ビ

ット(D9乃至D2)を使用して、MPEG規格にもとづいてデコードされたベースバンドのビデオデータを伝送するようにしている。このように、復号された8ビットのビデオデータをD1フォーマットに書き込むと、下位2ビット(D1とD0)は、空きビット(unallocated bits)となる。本発明のトランスコーディングシステム1ではこの空きエリア(unallocated area)を利用して、カウンタ値とともにヒストリ情報を伝送するようにしている。

【0085】この図7に記載されたデータブロックは、各サブブロック(Y[0], Y[1], Y[2], Y[3], Cr[0], Cr[1], Cb[0], Cb[1])における1ピクセルを伝送するためのデータブロックであるので、1マクロブロックのデータを伝送するためには、この図7に示されているデータブロックが64個伝送される。下位2ビット(D1とD0)を使用すれば、1マクロブロックのビデオデータに対して、合計で1024(=16×64)ビットのカウンタ値およびヒストリ情報を伝送できる。従って、1世代分のヒストリ情報は、256ビットとなるように生成されているので、過去の4(=1024/256)世代分のヒストリ情報を1マクロブロックのビデオデータに対して重畳することができる。図7に示した例では、第1世代のヒストリ情報、第2世代のヒストリ情報、および、第3世代のヒストリ情報、並びにカウンタ値が重畳されている。

【0086】また、輝度信号に対応するサブブロックに第1世代のヒストリ情報、第2世代のヒストリ情報、および、第3世代のヒストリ情報を重畳して、色差信号に対応するサブブロックにカウンタ値を重畳するようにしてもよい。

【0087】更に、16ビットのカウンタ値は、SMPTE(The Society of Motion Picture and Television Engineers) 291Mで規定されているAncillary Data Packetに格納して、伝送するようにしてもよい。図8は、Ancillary Data Packetの例を示す図である。

【0088】ADF(Ancillary Data Flag)には、Ancillary Data Packetの先頭を示す所定のデータ("000", "3FF", "3FF")が格納される。DID(Data Identification Word)には、Ancillary Data Packetに格納されるデータのフォーマットを特定する値、例えば、SMPTE 291Mで規定されているUser Applicationに対応する値(C0h乃至Cfhのいずれか)が格納される。

【0089】16ビットのカウンタ値を格納したワードは、3ワードに分割されてAncillary Data Packetに格納されるので、SDID(Secondary Data ID)に続くDC(Data Count Number Word)には、"3"が設定される。User Data Wordsには、図9に示すカウンタ値の下位6ビットを格納するワード(CC0)、図10に示すカウンタ値の7ビット乃至12ビットを格納するワード(CC1)、および図11に示すカウンタ値の上位4ビットを

格納するワード (CC2) が順に格納される。

【0090】図9は、カウンタ値の下位6ビットを格納するワード (CC0) の例を説明する図である。下位2ビット (B0とB1) は、それぞれ“0”が設定される。3ビット (B2) 乃至8ビット (B8) には、それぞれカウンタ値の下位6ビット (カウンタ値 [0] 乃至カウンタ値 [5]) が設定される。9ビットには、下位8ビット (B0乃至B7) の偶数パリティが設定され、10ビットには、9ビットに設定された偶数パリティの否定値が設定される。

【0091】図10は、カウンタ値の7ビット乃至12ビットを格納するワード (CC1) の例を説明する図である。下位2ビット (B0とB1) は、それぞれ“0”が設定される。3ビット (B2) 乃至8ビット (B8) には、それぞれカウンタ値の7ビット乃至12ビット (カウンタ値 [6] 乃至カウンタ値 [11]) が設定される。9ビットには、下位8ビット (B0乃至B7) の偶数パリティが設定され、10ビットには、9ビットに設定された偶数パリティの否定値が設定される。

【0092】図11は、カウンタ値の上位4ビットを格納するワード (CC2) の例を説明する図である。下位2ビット (B0とB1) は、それぞれ“0”が設定される。3ビット (B2) 乃至6ビット (B5) には、それぞれカウンタ値の上位4ビット (カウンタ値 [12] 乃至カウンタ値 [15]) が設定される。7ビット (B6) および8ビット (B7) は、それぞれ“0”が設定される。9ビットには、下位8ビット (B0乃至B7) の偶数パリティが設定され、10ビットには、9ビットに設定された偶数パリティの否定値が設定される。

【0093】Ancillary Data Packetの最後には、CS (Checksum Word) が格納される。

【0094】このように、トランスコーディングシステム1は、16ビットのカウンタ値をAncillary Data Packetに格納して伝送することができる。

【0095】カウンタ値は、ベースバンドデジタルビデオ信号のLSBに多重化されている符号化パラメータに多重化することもできる。図12は、ベースバンドデジタルビデオ信号のLSBに多重化されている符号化パラメータにカウンタ値を多重化するときの、ヒストリ情報多重化装置103およびカウンタ値多重化装置105に対応する機能の構成を説明する図である。タイミング信号発生装置271は、入力されたベースバンドデジタルビデオ信号を基に、ベースバンドデジタルビデオ信号のLSBに同期した、クロマタイミングパルスを生成し、スイッチ275に供給する。

【0096】カウンタ値フォーマット変換装置272は、カウンタ101から供給されたカウンタ値をユーザデータの方式に変換して、符号化パラメータフォーマット変換装置273に出力する。符号化パラメータフォーマット変換装置273は、復号装置102から供給され

た符号化パラメータ (3RD)、およびヒストリデコーディング装置104から供給された符号化パラメータ (1ST, 2ND) に、カウンタ値フォーマット変換装置272から供給されたユーザデータの方式のカウンタ値を多重化して、シリアル-パラレル変換装置274に出力する。

【0097】シリアル-パラレル変換装置274は、カウンタ値が多重化された符号化パラメータを、シリアルからパラレルに変換して、スイッチ275に供給する。スイッチ275は、タイミング信号発生回路271から供給されるクロマタイミングパルスを基に、ベースバンドデジタルビデオ信号のLSBにカウンタ値が多重化された符号化パラメータを多重化する。

【0098】このように、カウンタ値は、ベースバンドデジタルビデオ信号のブランキング部分に多重化されている符号化パラメータに多重化される。

【0099】また、カウンタ値は、ベースバンドデジタルビデオ信号の輝度または色差のブランキング部分に多重化されている符号化パラメータに多重化することもできる。図13は、ベースバンドデジタルビデオ信号の輝度または色差のブランキング部分に多重化されている符号化パラメータにカウンタ値を多重化するときの、ヒストリ情報多重化装置103およびカウンタ値多重化装置105に対応する機能の構成を説明する図である。タイミング信号発生回路281は、入力されたベースバンドデジタルビデオ信号を基に、ベースバンドデジタルビデオ信号の輝度または色差のブランキング部分に同期した、ブランキングタイミングパルスを生成し、スイッチ282に供給する。

【0100】カウンタ値フォーマット変換装置272は、カウンタ101から供給されたカウンタ値をユーザデータの方式に変換して、符号化パラメータフォーマット変換装置273に出力する。符号化パラメータフォーマット変換装置273は、復号装置102から供給された符号化パラメータ (3RD)、およびヒストリデコーディング装置104から供給された符号化パラメータ (1ST, 2ND) に、カウンタ値フォーマット変換装置272から供給されたユーザデータの方式のカウンタ値を多重化して、スイッチ282に出力する。

【0101】スイッチ282は、タイミング信号発生回路281から供給されるブランキングタイミングパルスを基に、ベースバンドデジタルビデオ信号の輝度または色差のブランキング部分にカウンタ値が多重化された符号化パラメータを多重化する。

【0102】このように、カウンタ値は、ベースバンドデジタルビデオ信号の輝度または色差のブランキング部分に多重化されている符号化パラメータに多重化される。

【0103】カウンタ値分離装置111は、D1フォーマットとして伝送されたデータの下位2ビットからカウンタ値を抽出するための回路である。カウンタ値分離装

10

20

30

40

50

置 111 は、カウンタ値が抽出され、分離された D1 フォーマットとして伝送されたデータをヒストリ情報分離装置 115 に供給する。

【0104】カウンタ値分離装置 111 は、D1 フォーマットとして伝送されたデータから分離されたカウンタ値をフォーマット変換装置 112 に供給するとともに、フレームまたはフィールドに同期した信号をカウンタ 113 に供給する。

【0105】カウンタ 113 は、カウンタ値分離装置 111 から供給されるフレームまたはフィールドに同期した信号を基に、カウントアップ（1 を加算）される 16 ビットのカウンタである。カウンタ 113 は、0 乃至 65,535 のいずれかのカウンタ値を比較装置 114 に出力する。

【0106】カウンタ 113 は、65,535 のカウンタ値を有する場合、カウンタ値分離装置 111 からフレームまたはフィールドに同期した信号が供給されたとき、カウンタ値を 0 とし、その後も、カウンタ値分離装置 111 から供給されるフレームまたはフィールドに同期した信号を基に、カウントアップを継続する。

【0107】なお、カウンタ 101 をカウントダウン（1 を減算）するようにしたとき、カウンタ 113 も、カウンタ値分離装置 111 から供給されるフレームまたはフィールドに同期した信号を基に、カウントダウンするように構成する。

【0108】フォーマット変換装置 112 は、カウンタ値分離装置 111 から供給された、D1 フォーマットとして伝送されたデータから分離されたカウンタ値の方式（例えば、図 9 乃至図 11 で説明したワードの方式）を 16 ビットのカウンタ値（例えば、カウンタ 113 が出力するカウンタ値と同じ方式）に変換して、比較装置 114 に出力する。

【0109】比較装置 114 は、フォーマット変換装置 112 から供給されたカウンタ値とカウンタ 113 から供給されたカウンタ値とを比較し、その値が同一であるとき、所定の値の不連続パラメータを符号化装置 116 に出力し、その値が異なるとき、他の値の不連続パラメータを符号化装置 116 に出力する。

【0110】比較装置 114 にフォーマット変換装置 112 から供給されたカウンタ値とカウンタ 113 から供給されたカウンタ値とが異なる値をとるとき、カウンタ 113 は、フォーマット変換装置 112 が出力するカウンタ値をロードして、自己のカウンタ値として設定する。このようにすることで、比較装置 114 は、一旦、画像の不連続点を検出した後でも、再度、画像に不連続点が含まれていた場合、その画像の不連続点を検出することかできる。

【0111】ベースバンドデジタルビデオ信号に含まれる画像が編集等によりつなぎ合わされた、所定のフレームが挿入された、またはフレームが削除されるとき、画像

に対応してベースバンドデジタルビデオ信号に格納されたカウンタ値は不連続の値をとるので、フォーマット変換装置 112 から供給されたカウンタ値とカウンタ 113 から供給されたカウンタ値とは異なる値となり、比較装置 114 は、他の値の不連続パラメータを符号化装置 116 に供給する。ベースバンドデジタルビデオ信号に含まれる画像が編集等されていない場合、画像に対応してベースバンドデジタルビデオ信号に格納されたカウンタ値は連続した値をとるので、フォーマット変換装置 112 から供給されたカウンタ値とカウンタ 113 から供給されたカウンタ値とは同じ値となり、比較装置 114 は、所定の値の不連続パラメータを符号化装置 116 に供給する。

【0112】このように、符号化装置 116 は、比較装置 114 から供給される信号を基に、画像が編集等されたか否かを判定することができる。

【0113】ヒストリ情報分離装置 115 は、D1 フォーマットとして伝送されたデータの上位 8 ビットから、ベースバンドビデオデータを抽出し、下位 2 ビットからヒストリ情報を抽出するための回路である。図 4 に示した例では、ヒストリ情報分離装置 115 は、伝送データからベースバンドのビデオデータを抽出して、そのビデオデータを符号化装置 116 に供給するとともに、伝送データから第 1 世代、第 2 世代及び第 3 世代のヒストリ情報を抽出して、符号化装置 116 とヒストリエンコーディング装置 117 にそれぞれ供給する。

【0114】符号化装置 116 は、ヒストリ情報分離装置 115 から供給されたベースバンドのビデオデータを、オペレータまたはホストコンピュータから指定された GOP 構造及びビットレートを有するビットストリームになるように符号化するための装置である。なお、GOP 構造を変更するとは、たとえば、GOP に含まれるピクチャの数、I ピクチャと I ピクチャの間に存在する P ピクチャの数、及び I ピクチャと P ピクチャ（または I ピクチャ）の間に存在する B ピクチャの数を変更することを意味する。

【0115】図 4 に示された例では、供給されたベースバンドのビデオデータには、第 1 世代、第 2 世代及び第 3 世代のヒストリ情報が重畳されているので、この符号化装置 116 は、再符号化処理による画質劣化が少なくなるように、これらのヒストリ情報を選択的に再利用して第 4 世代の符号化処理を行う。

【0116】図 14 は、この符号化装置 116 に設けられているエンコーダ 301 の具体的な構成を示している図である。このエンコーダ 301 は、動きベクトル検出回路 310、フレームメモリ 311、フレーム／フィールド予測モード切り替え回路 312、演算器 313、DCT モード切り替え回路 315、DCT 回路 316、量子化回路 317、可変長符号化回路 318、伝送バッファ 319、逆量子化回路 320、逆 DCT 回路 321、

10

20

30

40

50

演算器 322、フレームメモリ 323、並びに動き補償回路 324 を備えている。

【0117】始めに、履歴情報が無い場合のエンコーダ 301 の参照ピクチャの符号化処理を説明する。

【0118】符号化される画像データは、マクロブロック単位で動きベクトル検出回路 310 に入力される。動きベクトル検出回路 310 は、予め設定されている所定のシーケンスに従って、各フレームの画像データを、I ピクチャ、P ピクチャ、または B ピクチャとして処理する。シーケンシャルに入力される各フレームの画像を、I、P、または B のいずれのピクチャとして処理するかは、予め定められている（例えば、フレーム F1 乃至 F17 により構成されるグループオブピクチャが、I、B、P、B、P、・・・B、P として処理される）。

【0119】I ピクチャとして処理されるフレーム（例えば、フレーム F1）の画像データは、動きベクトル検出回路 310 からフレームメモリ 311 の前方原画像部 311a に転送、記憶され、B ピクチャとして処理されるフレーム（例えば、フレーム F2）の画像データは、参照原画像部 311b に転送、記憶され、P ピクチャとして処理されるフレーム（例えば、フレーム F3）の画像データは、後方原画像部 311c に転送、記憶される。

【0120】また、次のタイミングにおいて、さらに B ピクチャ（フレーム F4）または P ピクチャ（フレーム F5）として処理すべきフレームの画像が入力されたとき、それまで後方原画像部 311c に記憶されていた最初の P ピクチャ（フレーム F3）の画像データが、前方原画像部 311a に転送され、次の B ピクチャ（フレーム F4）の画像データが、参照原画像部 311b に記憶（上書き）され、次の P ピクチャ（フレーム F5）の画像データが、後方原画像部 311c に記憶（上書き）される。このような動作が順次繰り返される。

【0121】フレームメモリ 311 に記憶された各ピクチャの信号は、そこから読み出され、Frame/Field 予測モード切り替え回路 312 において、フレーム予測モード処理、またはフィールド予測モード処理が行なわれる。

【0122】さらにまた、コントローラ 330 の制御の下に、演算器 313 において、画像内予測、前方予測、後方予測、または両方向予測の演算が行なわれる。これらの処理のうち、いずれの処理を行なうかは、予測誤差信号（処理の対象とされている参照画像と、これに対する予測画像との差分）に対応して決定される。このため、動きベクトル検出回路 310 は、この判定に用いられる予測誤差信号の絶対値和（自乗和でもよい）を生成する。

【0123】ここで、Frame/Field 予測モード切り替え回路 312 におけるフレーム予測モードとフィールド予測モードについて説明する。

【0124】フレーム予測モードが設定された場合においては、Frame/Field 予測モード切り替え回路 312

は、動きベクトル検出回路 310 より供給される 4 個の輝度ブロック Y[1] 乃至 Y[4] を、そのまま後段の演算器 313 に出力する。すなわち、この場合においては、各輝度ブロックに奇数フィールドのラインのデータと、偶数フィールドのラインのデータとが混在した状態となっている。このフレーム予測モードにおいては、4 個の輝度ブロック（マクロブロック）を単位として予測が行われ、4 個の輝度ブロックに対して 1 個の動きベクトルが対応される。

【0125】これに対して、Frame/Field 予測モード切り替え回路 312 は、フィールド予測モードにおいては、動きベクトル検出回路 310 より入力される信号を、4 個の輝度ブロックのうち、輝度ブロック Y[1] と Y[2] を、例えば奇数フィールドのラインのドットだけで構成させ、他の 2 個の輝度ブロック Y[3] と Y[4] を、偶数フィールドのラインのドットだけで構成させて、演算器 313 に出力する。この場合においては、2 個の輝度ブロック Y[1] と Y[2] に対して、1 個の動きベクトルが対応され、他の 2 個の輝度ブロック Y[3] と Y[4] に対して、他の 1 個の動きベクトルが対応される。

【0126】動きベクトル検出回路 310 は、フレーム予測モードにおける予測誤差の絶対値和、およびフィールド予測モードにおける予測誤差の絶対値和を Frame/Field 予測モード切り替え回路 312 に出力する。Frame/Field 予測モード切り替え回路 312 は、フレーム予測モードとフィールド予測モードにおける予測誤差の絶対値和を比較し、その値が小さい予測モードに対応する処理を施して、データを演算器 313 に出力する。

【0127】ただし、このような処理は、実際には動きベクトル検出回路 310 で行われる。すなわち、動きベクトル検出回路 310 は、決定されたモードに対応する構成の信号を Frame/Field 予測モード切り替え回路 312 に出力し、Frame/Field 予測モード切り替え回路 312 は、その信号を、そのまま後段の演算器 313 に出力する。

【0128】なお、色差信号は、フレーム予測モードの場合、奇数フィールドのラインのデータと偶数フィールドのラインのデータとが混在する状態で、演算器 313 に供給される。また、フィールド予測モードの場合、各色差ブロック Cb、Cr の上半分（4 ライン）が、輝度ブロック Y[1]、Y[2] に対応する奇数フィールドの色差信号とされ、下半分（4 ライン）が、輝度ブロック Y[3]、Y[4] に対応する偶数フィールドの色差信号とされる。

【0129】また、動きベクトル検出回路 310 は、以下に示すようにして、コントローラ 330 において、画像内予測、前方予測、後方予測、または両方向予測のい

ずれの予測を行なうかを決定するための予測誤差の絶対値和を生成する。

【0130】すなわち、画像内予測の予測誤差の絶対値和として、参照画像のマクロブロックの信号 $A_{ij}$ の総和 $\Sigma A_{ij}$ の絶対値 $|\Sigma A_{ij}|$ と、マクロブロックの信号 $A_{ij}$ の絶対値 $|A_{ij}|$ の総和 $\Sigma |A_{ij}|$ の差を求める。また、前方予測の予測誤差の絶対値和として、参照画像のマクロブロックの信号 $A_{ij}$ と、予測画像のマクロブロックの信号 $B_{ij}$ の差 $A_{ij}-B_{ij}$ の絶対値 $|A_{ij}-B_{ij}|$ の総和 $\Sigma |A_{ij}-B_{ij}|$ を求める。また、後方予測と両方向予測の予測誤差の絶対値和も、前方予測における場合と同様に（その予測画像を前方予測における場合と異なる予測画像に変更して）求める。

【0131】これらの絶対値和は、コントローラ330に供給される。コントローラ330は、前方予測、後方予測および両方向予測の予測誤差の絶対値和のうちの最も小さいものを、インタ予測の予測誤差の絶対値和として選択する。さらに、このインタ予測の予測誤差の絶対値和と、画像内予測の予測誤差の絶対値和とを比較し、その小さい方を選択し、この選択した絶対値和に対応するモードを予測モードとして選択する。すなわち、画像内予測の予測誤差の絶対値和の方が小さければ、画像内予測モードが設定される。インタ予測の予測誤差の絶対値和の方が小さければ、前方予測、後方予測または両方向予測モードのうちの対応する絶対値和が最も小さかったモードが設定される。

【0132】このように、動きベクトル検出回路310は、参照画像のマクロブロックの信号を、フレームまたはフィールド予測モードのうち、Frame/Field予測モード切り替え回路312により選択されたモードに対応する構成で、Frame/Field予測モード切り替え回路312を介して演算器313に供給するとともに、4つの予測モードのうちのコントローラ330により選択された予測モードに対応する予測画像と参照画像の間の動きベクトルを検出し、可変長符号化回路318と動き補償回路324に出力する。上述したように、この動きベクトルとしては、対応する予測誤差の絶対値和が最小となるものが選択される。

【0133】コントローラ330は、動きベクトル検出回路310が前方原画像部311aより1ピクチャの画像データを読み出しているとき、予測モードとして、フレームまたはフィールド（画像）内予測モード（動き補償を行わないモード）を設定し、演算器313のスイッチ313dを接点a側に切り替える。これにより、1ピクチャの画像データがFrame/FieldDCTモード切り替え回路315に入力される。

【0134】Frame/FieldDCTモード切り替え回路315は、4個の輝度ブロックのデータを、奇数フィールドのラインと偶数フィールドのラインが混在する状態（フレームDCTモード）、または、分離された状態（フィール

ドDCTモード）、のいずれかの状態にして、DCT回路316に出力する。

【0135】すなわち、Frame/FieldDCTモード切り替え回路315は、奇数フィールドと偶数フィールドのデータを混在してDCT処理した場合における符号化効率と、分離した状態においてDCT処理した場合の符号化効率とを比較し、符号化効率の良好なモードを選択する。

【0136】例えば、入力された信号を、奇数フィールドと偶数フィールドのラインが混在する構成とし、上下に隣接する奇数フィールドのラインの信号と偶数フィールドのラインの信号の差を演算し、さらにその絶対値の和（または自乗和）を求める。

【0137】また、入力された信号を、奇数フィールドと偶数フィールドのラインが分離した構成とし、上下に隣接する奇数フィールドのライン同士の信号の差と、偶数フィールドのライン同士の信号の差を演算し、それぞれの絶対値の和（または自乗和）を求める。

【0138】さらに、両者（絶対値和）を比較し、小さい値に対応するDCTモードを設定する。すなわち、前者の方が小さければ、フレームDCTモードを設定し、後者の方が小さければ、フィールドDCTモードを設定する。

【0139】そして、選択したDCTモードに対応する構成のデータをDCT回路316に出力するとともに、選択したDCTモードを示すDCTフラグを、可変長符号化回路318、および動き補償回路324に出力する。

【0140】Frame/Field予測モード切り替え回路312における予測モードと、このFrame/FieldDCTモード切り替え回路315におけるDCTモードを比較して明らかに、輝度ブロックに関しては、両者の各モードにおけるデータ構造は実質的に同一である。

【0141】Frame/Field予測モード切り替え回路312において、フレーム予測モード（奇数ラインと偶数ラインが混在するモード）が選択された場合、Frame/FieldDCTモード切り替え回路315においても、フレームDCTモード（奇数ラインと偶数ラインが混在するモード）が選択される可能性が高く、またFrame/Field予測モード切り替え回路312において、フィールド予測モード（奇数フィールドと偶数フィールドのデータが分離されたモード）が選択された場合、Frame/FieldDCTモード切り替え回路315において、フィールドDCTモード（奇数フィールドと偶数フィールドのデータが分離されたモード）が選択される可能性が高い。

【0142】しかしながら、必ずしも常にこのようにモードが選択されるわけではなく、Frame/Field予測モード切り替え回路312においては、予測誤差の絶対値和が小さくなるようにモードが決定され、Frame/FieldDCTモード切り替え回路315においては、符号化効率が良好となるようにモードが決定される。

【0143】Frame/FieldDCTモード切り替え回路315より出力された1ピクチャの画像データは、DCT回路3

10

20

30

40

50

16に入力されてDCT処理され、DCT係数に変換される。このDCT係数は、量子化回路317に入力され、送信バッファ319のデータ蓄積量（バッファ蓄積量）に対応した量子化スケールで量子化された後、可変長符号化回路318に入力される。

【0144】可変長符号化回路318は、量子化回路317より供給される量子化スケール（スケール）に対応して、量子化回路317より供給される画像データ（いまの場合、Iピクチャのデータ）を、例えばハフマン符号などの可変長符号に変換し、送信バッファ319に出力する。

【0145】可変長符号化回路318にはまた、量子化回路317より量子化スケール（スケール）、コントローラ330より予測モード（画像内予測、前方予測、後方予測、または両方向予測のいずれが設定されたかを示すモード）、動きベクトル検出回路310より動きベクトル、Frame/Field予測モード切り替え回路312より予測フラグ（フレーム予測モードまたはフィールド予測モードのいずれが設定されたかを示すフラグ）、およびFrame/FieldDCTモード切り替え回路315が出力するDCTフラグ（フレームDCTモードまたはフィールドDCTモードのいずれが設定されたかを示すフラグ）が入力されており、これらも可変長符号化される。

【0146】送信バッファ319は、入力されたデータを一時蓄積し、蓄積量に対応するデータを量子化回路317に出力する。送信バッファ319は、そのデータ残量が許容上限値まで増量すると、量子化制御信号によって量子化回路317の量子化スケールを大きくすることにより、量子化データのデータ量を低下させる。また、これとは逆に、データ残量が許容下限値まで減少すると、送信バッファ319は、量子化制御信号によって量子化回路317の量子化スケールを小さくすることにより、量子化データのデータ量を増大させる。このようにして、送信バッファ319のオーバーフローまたはアンダフローが防止される。

【0147】そして、送信バッファ319に蓄積されたデータは、所定のタイミングで読み出され、伝送路に出力される。

【0148】一方、量子化回路317より出力されたIピクチャのデータは、逆量子化回路320に入力され、量子化回路317より供給される量子化スケールに対応して逆量子化される。逆量子化回路320の出力は、IDCT（逆離散コサイン変換）回路321に入力され、逆離散コサイン変換処理された後、演算器322を介してフレームメモリ323の前方予測画像部323a供給されて記憶される。

【0149】動きベクトル検出回路310は、シーケンシャルに入力される各フレームの画像データを、たとえば、I、B、P、B、P、B・・・のピクチャとしてそれぞれ処理する場合、最初に入力されたフレームの画像

データをIピクチャとして処理した後、次に入力されたフレームの画像をBピクチャとして処理する前に、さらにその次に入力されたフレームの画像データをPピクチャとして処理する。Bピクチャは、後方予測を伴うため、後方予測画像としてのPピクチャが先に用意されていないと、復号することができないからである。

【0150】そこで動きベクトル検出回路310は、Iピクチャの処理の次に、後方原画像部311cに記憶されているPピクチャの画像データの処理を開始する。そして、上述した場合と同様に、マクロブロック単位でのフレーム間差分（予測誤差）の絶対値和が、動きベクトル検出回路310からFrame/Field予測モード切り替え回路312とコントローラ330に供給される。Frame/Field予測モード切り替え回路312とコントローラ330は、このPピクチャのマクロブロックの予測誤差の絶対値和に対応して、フレーム／フィールド予測モード、または画像内予測、前方予測、後方予測、もしくは両方向予測の予測モードを設定する。

【0151】演算器313は、画像内予測モードが設定されたとき、スイッチ313dを上述したように接点a側に切り替える。したがって、このデータは、Iピクチャのデータと同様に、Frame/FieldDCTモード切り替え回路315、DCT回路316、量子化回路317、可変長符号化回路318、および送信バッファ319を介して伝送路に伝送される。また、このデータは、逆量子化回路320、IDCT回路321、および演算器322を介してフレームメモリ323の後方予測画像部323bに供給されて記憶される。

【0152】また、前方予測モードが設定された場合、スイッチ313dが接点bに切り替えられるとともに、フレームメモリ323の前方予測画像部323aに記憶されている画像（いまの場合、Iピクチャの画像）データが読み出され、動き補償回路324により、動きベクトル検出回路310が出力する動きベクトルに対応して動き補償される。すなわち、動き補償回路324は、コントローラ330より前方予測モードの設定が指令されたとき、前方予測画像部323aの読み出しアドレスを、動きベクトル検出回路310が、現在、出力しているマクロブロックの位置に対応する位置から動きベクトルに対応する分だけずらしてデータを読み出し、予測画像データを生成する。

【0153】動き補償回路324より出力された予測画像データは、演算器313aに供給される。演算器313aは、Frame/Field予測モード切り替え回路312より供給された参照画像のマクロブロックのデータから、動き補償回路324より供給された、このマクロブロックに対応する予測画像データを減算し、その差分（予測誤差）を出力する。この差分データは、Frame/FieldDCTモード切り替え回路315、DCT回路316、量子化回路317、可変長符号化回路318、および送信バッファ

ァ 319 を介して伝送路に伝送される。また、この差分データは、逆量子化回路 320、および IDCT 回路 321 により局所的に復号され、演算器 322 に入力される。

【0154】この演算器 322 にはまた、演算器 313 a に供給されている予測画像データと同一のデータが供給されている。演算器 322 は、IDCT 回路 321 が出力する差分データに、動き補償回路 324 が出力する予測画像データを加算する。これにより、元の（復号した）P ピクチャの画像データが得られる。この P ピクチャの画像データは、フレームメモリ 323 の後方予測画像部 323 b に供給されて記憶される。

【0155】動きベクトル検出回路 310 は、このように、I ピクチャと P ピクチャのデータが前方予測画像部 323 a と後方予測画像部 323 b にそれぞれ記憶された後、次に B ピクチャの処理を実行する。Frame/Field 予測モード切り替え回路 312 とコントローラ 330 は、マクロブロック単位でのフレーム間差分の絶対値和の大きさに対応して、フレーム/フィールドモードを設定し、また、予測モードを画像内予測モード、前方予測モード、後方予測モード、または両方向予測モードのいずれかに設定する。

【0156】上述したように、画像内予測モードまたは前方予測モードの時、スイッチ 313 d は接点 a または b に切り替えられる。このとき、P ピクチャにおける場合と同様の処理が行われ、データが伝送される。

【0157】これに対して、後方予測モードまたは両方向予測モードが設定された時、スイッチ 313 d は、接点 c または d にそれぞれ切り替えられる。

【0158】スイッチ 313 d が接点 c に切り替えられている後方予測モードの時、後方予測画像部 323 b に記憶されている画像（いまの場合、P ピクチャの画像）データが読み出され、動き補償回路 324 により、動きベクトル検出回路 310 が出力する動きベクトルに対応して動き補償される。すなわち、動き補償回路 324 は、コントローラ 330 より後方予測モードの設定が指令されたとき、後方予測画像部 323 b の読み出しアドレスを、動きベクトル検出回路 310 が、現在、出力しているマクロブロックの位置に対応する位置から動きベクトルに対応する分だけずらしてデータを読み出し、予測画像データを生成する。

【0159】動き補償回路 324 より出力された予測画像データは、演算器 313 b に供給される。演算器 313 b は、Frame/Field 予測モード切り替え回路 312 より供給された参照画像のマクロブロックのデータから、動き補償回路 324 より供給された予測画像データを減算し、その差分を出力する。この差分データは、Frame/Field DCT モード切り替え回路 315、DCT 回路 316、量子化回路 317、可変長符号化回路 318、および送信バッファ 319 を介して伝送路に伝送される。

【0160】スイッチ 313 d が接点 d に切り替えられ

ている両方向予測モードの時、前方予測画像部 323 a に記憶されている画像（いまの場合、I ピクチャの画像）データと、後方予測画像部 323 b に記憶されている画像（いまの場合、P ピクチャの画像）データが読み出され、動き補償回路 324 により、動きベクトル検出回路 310 が出力する動きベクトルに対応して動き補償される。

【0161】すなわち、動き補償回路 324 は、コントローラ 330 より両方向予測モードの設定が指令されたとき、前方予測画像部 323 a と後方予測画像部 323 b の読み出しアドレスを、動きベクトル検出回路 310 がいま出力しているマクロブロックの位置に対応する位置から動きベクトル（この場合の動きベクトルは、前方予測画像用と後方予測画像用の 2 つとなる）に対応する分だけずらしてデータを読み出し、予測画像データを生成する。

【0162】動き補償回路 324 より出力された予測画像データは、演算器 313 c に供給される。演算器 313 c は、動きベクトル検出回路 310 より供給された参照画像のマクロブロックのデータから、動き補償回路 324 より供給された予測画像データの平均値を減算し、その差分を出力する。この差分データは、Frame/Field DCT モード切り替え回路 315、DCT 回路 316、量子化回路 317、可変長符号化回路 318、および送信バッファ 319 を介して伝送路に伝送される。

【0163】B ピクチャの画像は、他の画像の予測画像とされることがないため、フレームメモリ 323 には記憶されない。

【0164】なお、フレームメモリ 323 において、前方予測画像部 323 a と後方予測画像部 323 b は、必要に応じてバンク切り替えが行われ、所定の参照画像に対して、一方または他方に記憶されているものを、前方予測画像あるいは後方予測画像として切り替えて出力することができる。

【0165】上述した説明においては、輝度ブロックを中心として説明をしたが、色差ブロックについても同様に、マクロブロックを単位として処理されて伝送される。なお、色差ブロックを処理する場合の動きベクトルは、対応する輝度ブロックの動きベクトルを垂直方向と水平方向に、それぞれ 1/2 にしたものが用いられる。

【0166】さらに、コントローラ 330 は、オペレータまたはホストコンピュータから GOP 構造に関するインストラクションを受け取って、その GOP 構造に対応するように各ピクチャのピクチャタイプを決定する。また、このコントローラ 330 は、オペレータまたはホストコンピュータからターゲットビットレートの情報を受け取り、このエンコーダ 301 から出力されるビットレートがこの指定されたターゲットビットレートになるように、量子化回路 317 を制御する。

【0167】さらに、このコントローラ 330 は、ヒス

トリ情報分離装置 1 1 5 から出力された複数世代のヒストリ情報を受け取り、これらのヒストリ情報を再利用して参照ピクチャの符号化処理を行う。以下に詳しく説明する。

【0 1 6 8】まず、このコントローラ 3 3 0 は、オペレータによって指定された GOP 構造から決定された参照ピクチャのピクチャタイプと、ヒストリ情報に含まれるピクチャタイプが一致するか否かを判断する。つまり、指定されたピクチャタイプと同じピクチャタイプでこの参照ピクチャが過去において符号化されたことがあるか否かを判断する。

【0 1 6 9】図 4 に示された例をあげてよりわかりやすく説明するのであれば、このコントローラ 3 3 0 は、第 4 世代の符号化処理としてこの参照ピクチャにアサインされたピクチャタイプが、第 1 世代の符号化処理におけるこの参照ピクチャのピクチャタイプ、第 2 世代の符号化処理におけるこの参照ピクチャのピクチャタイプ、または第 3 世代の符号化処理におけるこの参照ピクチャのピクチャタイプのいずれかと一致するか否かを判断する。

【0 1 7 0】もし、第 4 世代の符号化処理としてこの参照ピクチャに指定されたピクチャタイプが、過去の符号化処理におけるどのピクチャタイプとも一致しないのであれば、このコントローラ 3 3 0 は、上述した「通常符号化処理」を行う。つまり、この場合には、第 1 世代、第 2 世代または第 3 世代のどの世代の符号化処理においても、第 4 世代の符号化処理としてアサインされたピクチャタイプで、この参照ピクチャが符号化処理されたことがないということになる。一方、もし、第 4 世代の符号化処理としてこの参照ピクチャに指定されたピクチャタイプが、過去の符号化処理におけるいずれかのピクチャタイプと一致するのであれば、このコントローラ 3 3 0 は、「パラメータ再利用符号化処理」を行う。つまり、この場合には、第 1 世代、第 2 世代または第 3 世代のいずれかの世代の符号化処理において、第 4 世代の符号化処理としてアサインされたピクチャタイプで、この参照ピクチャが符号化処理されたことがあるということになる。

【0 1 7 1】また、「パラメータ再利用符号化処理」を実行している場合であっても、比較装置 1 1 4 から供給される不連続パラメータが、画像が不連続であること（ベースバンドデジタルビデオ信号に含まれる画像が編集等によりつなぎ合わされた、所定のフレームが挿入された、またはフレームが削除されたなど）を示す値を有するとき、符号化パラメータを再利用して符号化処理を行うと、不連続な点の前後の画像が大きく劣化するので、コントローラ 3 3 0 は、「通常符号化処理」を行う。

【0 1 7 2】ヒストリ情報が無い場合のエンコーダ 3 0 1 の参照ピクチャの符号化処理と、一部、説明が重複するが、まず、最初にコントローラ 3 3 0 の通常符号化処

理について説明する。

【0 1 7 3】動きベクトル検出回路 3 1 0 は、フレーム予測モードまたはフィールド予測モードのどちらが選択されるべきかを判断するために、フレーム予測モードにおける予測誤差とフィールド予測モードにおける予測誤差をそれぞれ検出し、その予測誤差の値をコントローラ 3 3 0 に供給する。コントローラ 3 3 0 は、それらの予測誤差の値を比較し、その予測誤差の値が小さい方の予測モードを選択する。Frame/Field 予測モード切り替え回路 3 1 2 は、コントローラ 3 3 0 によって選択された予測モードに対応するように信号処理を行い、それを演算器 3 1 3 に供給する。

【0 1 7 4】具体的には、Frame/Field 予測モード切り替え回路 3 1 2 は、フレーム予測モードが選択された場合には、輝度信号に関しては、入力された状態のまま演算器 3 1 3 に出力するように信号処理を行い、色差信号に関しては、奇数フィールドラインと偶数フィールドラインとが混在するように信号処理する。一方、フィールド予測モードが選択された場合には、輝度信号に関しては、輝度ブロック Y [1] と Y [2] を奇数フィールドラインで構成し、輝度ブロック Y [3] と Y [4] を偶数フィールドラインで構成するように信号処理し、色差信号に関しては、上 4 ラインを奇数フィールドラインで構成し、下 4 ラインを偶数フィールドラインで構成するように信号処理する。

【0 1 7 5】さらに、動きベクトル検出回路 3 1 0 は、画像内予測モード、前方予測モード、後方予測モード、または両方向予測モードのうちのいずれの予測モードを選択するかを決定するために、各予測モードにおける予測誤差を生成し、各予測モードにおける予測誤差をコントローラ 3 3 0 にそれぞれ供給する。コントローラ 3 3 0 は、前方予測、後方予測および両方向予測の予測誤差のうちの最も小さいものを、インタ予測の予測誤差として選択する。さらに、このインタ予測の予測誤差と、画像内予測の予測誤差とを比較し、その小さい方を選択し、この選択した予測誤差に対応するモードを予測モードとして選択する。すなわち、画像内予測の予測誤差の方が小さければ、画像内予測モードが設定される。インタ予測の予測誤差の方が小さければ、前方予測、後方予測または両方向予測モードのうちの対応する予測誤差が最も小さかったモードが設定される。コントローラ 3 3 0 は、選択した予測モードに対応するように、演算器 3 1 3 及び動き補償回路 3 2 4 を制御する。

【0 1 7 6】DCT モード切り替え回路 3 1 5 は、フレーム DCT モードまたはフィールド DCT モードのいずれかを選択するために、4 個の輝度ブロックのデータを、奇数フィールドラインと偶数フィールドラインが混在するような信号形態（フレーム DCT モード）に変換するとともに、奇数フィールドラインと偶数フィールドラインが分離された信号形態（フィールド DCT モード）に変換し



て、それぞれの信号をDCT回路316に供給する。DCT回路316は、奇数フィールドと偶数フィールドを混在してDCT処理した場合における符号化効率と、奇数フィールドと偶数フィールドを分離した状態においてDCT処理した場合の符号化効率を計算し、その結果をコントローラ330に供給する。コントローラ330は、DCT回路316から供給されたそれぞれの符号化効率を比較し、符号化効率の良い方のDCTモードを選択し、その選択したDCTモードとなるようにDCTモード切り替え回路315を制御する。

【0177】コントローラ330は、オペレータまたはホストコンピュータから供給された目標ビットレートを示すターゲットビットレートと、送信バッファ319にバッファリングされているビット量を示す信号、つまり、バッファ残量を示す信号を受け取り、このターゲットビットレートとバッファ残量に基づいて、量子化回路317の量子化ステップサイズをコントロールするためのfeedback\_q\_scale\_codeを生成する。このfeedback\_q\_scale\_codeは、この送信バッファ319がオーバーフローまたはアンダーフローしないように、この送信バッファ319のバッファ残量に応じて生成される制御信号であって、また、送信バッファ319から出力されるビットストリームのビットレートが、ターゲットビットレートになるように制御する信号でもある。

【0178】具体的には、例えば、送信バッファ319にバッファリングされているビット量が少なくなってしまう場合には、次に符号化するピクチャの発生ビット量が増えるように、量子化ステップサイズを小さくし、一方、送信バッファ319にバッファリングされているビット量が多くなってしまった場合には、次に符号化するピクチャの発生ビット量が少なくなるように、量子化ステップサイズを大きくする。なお、feedback\_q\_scale\_codeと量子化ステップサイズは比例し、feedback\_q\_scale\_codeを大きくすると、量子化ステップサイズは大きくなり、feedback\_q\_scale\_codeを小さくすると、量子化ステップサイズは小さくなる。

【0179】次に、このトランスコーディングシステム1の特徴の1つでもある、パラメータ再利用符号化処理について説明する。この処理をより分かりやすく説明するために、参照ピクチャは、第1世代の符号化処理においてPピクチャとして符号化され、第2世代の符号化処理においてIピクチャとして符号化処理され、第3世代の符号化処理においてBピクチャとして符号化されていたものとし、今回の第4世代の符号化処理において、この参照ピクチャをPピクチャとして符号化しなければならないものとする。

【0180】この場合には、第4世代のピクチャタイプとしてアサインされたピクチャタイプと同じピクチャタイプ（Iピクチャ）で、この参照ピクチャは第1世代の符号化処理において符号化されているので、コントロー

ラ330は、供給されたビデオデータから符号化パラメータを新しく作成するのではなく、第1世代の符号化パラメータを使用して符号化処理を行う。この第4の符号化処理において再利用する符号化パラメータは、代表的なパラメータとしては、量子化スケールステップサイズを示すquantiser\_scale\_code、予測方向モードを示すmacroblock\_type、動きベクトルを示すmotion\_vector、Frame予測モードかField予測モードかを示すframe/field\_motion\_type、及びFrameDCTモードかFieldDCTモードかを示すdct\_type等である。

【0181】コントローラ330は、ヒストリ情報として伝送されたすべての符号化パラメータを再利用するわけではなく、再利用した方が望ましいと想定される上述したような符号化パラメータについては再利用し、再利用しない方が望ましいと考えられる符号化パラメータについては、新しく生成する。

【0182】次に、第4世代の符号化処理として、この参照ピクチャに指定されたピクチャタイプが、過去の符号化処理におけるいずれかのピクチャタイプと一致し、かつ、比較装置114から供給される不連続パラメータが、画像が連続であることを示す値を有するとき実行される、符号化パラメータ再利用符号化処理について、上述した通常符号化処理と異なる点を中心に説明する。

【0183】動きベクトル検出回路310は、上述した通常符号化処理においては、参照ピクチャの動きベクトルの検出を行ったが、このパラメータ再利用符号化処理においては、動きベクトルmotion\_vectorの検出処理は行わずに、第1世代のヒストリ情報として供給された動きベクトルmotion\_vectorを再利用する。その理由について説明する。

【0184】第3世代の符号化ストリームを復号したベースバンドのビデオデータは、少なくとも3回の復号及び符号化処理が行われているので、オリジナルビデオデータに比べると、明らかに画質が劣化している。画質が劣化しているビデオデータから動きベクトルを検出したとしても、正確な動きベクトルは検出できない。つまり、第4世代の符号化処理において検出された動きベクトルよりも、第1世代のヒストリ情報として供給されている動きベクトルの方が、明らかに、精度の高い動きベクトルである。つまり、第1世代の符号化パラメータとして伝送された動きベクトルを再利用することによって、第4世代の符号化処理を行ったとしても画質が劣化しない。コントローラ330は、この第1世代のヒストリ情報として供給された動きベクトルmotion\_vectorを、第4世代の符号化処理において符号化されるこの参照ピクチャの動きベクトル情報として、動き補償回路324及び可変長符号化回路318に供給する。

【0185】さらに、動きベクトル検出回路310は、フレーム予測モードとフィールド予測モードのどちらが選択されるかを判断するために、フレーム予測モードに

10

20

30

40

50

における予測誤差とフィールド予測モードにおける予測誤差をそれぞれ検出したが、このパラメータ再利用符号化処理においては、このフレーム予測モードにおける予測誤差とフィールド予測モードにおける予測誤差を検出する処理は行わずに、第1世代のヒストリ情報として供給されているFrame予測モードかField予測モードかを示すframe/field\_motion\_typeを再利用する。なぜなら、第4世代の符号化処理において検出された各予測モードにおける予測誤差よりも、第1世代において検出された各予測モードにおける予測誤差の方が精度が高いため、精度の高い予測誤差によって決定された予測モードを選択した方がより最適な符号化処理を行うことができるからである。

【0186】具体的には、コントローラ330は、この第1世代のヒストリ情報として供給されているframe/field\_motion\_typeに対応する制御信号をFrame/Field予測モード切り替え回路312に供給し、Frame/Field予測モード切り替え回路312は、この再利用されたframe/field\_motion\_typeに対応した信号処理を行う。

【0187】さらには、動きベクトル検出回路310は、通常符号化処理においては、画像内予測モード、前方予測モード、後方予測モード、または両方向予測モードのうちのいずれの予測モード（以下、この予測モードを、予測方向モードとも称する）を選択するかを決定するために、各予測方向モードにおける予測誤差を計算していたが、このパラメータ再利用符号化処理においては、各予測方向モードにおける予測誤差の計算は行わず、第1世代のヒストリ情報として供給されたmacroblock\_typeに基づいて予測方向モードを決定する。なぜなら、第4世代の符号化処理における各予測方向モードにおける予測誤差よりも、第1世代の符号化処理における各予測方向モードにおける予測誤差の方がより精度が高いため、より精度の高い予測誤差によって決定された予測方向モードを選択した方が、より効率の良い符号化処理が行えるからである。具体的には、コントローラ330は、第1世代のヒストリ情報に含まれているmacroblock\_typeによって示される予測方向モードを選択し、その選択した予測方向モードに対応するように、演算器313及び動き補償回路324をコントロールする。

【0188】DCTモード切り替え回路315は、通常符号化処理においては、フレームDCTモードの符号化効率と、フィールドDCTモードの符号化効率を比較するために、フレームDCTモードの信号形態に変換した信号と、フィールドDCTモードの信号形態に変換した信号の両方をDCT回路316に供給していたが、このパラメータ再利用符号化処理では、フレームDCTモードの信号形態に変換した信号と、フィールドDCTモードの信号形態に変換した信号の両方を生成する処理は行っておらず、第1世代のヒストリ情報に含まれているdct\_typeによって示されたDCTモードに対応する処理のみを行

っている。具体的には、コントローラ330は、第1世代のヒストリ情報に含まれているdct\_typeを再利用し、DCTモード切り替え回路315がこのdct\_typeによって示されるDCTモードに対応した信号処理を行うように、DCTモード切り替え回路315をコントロールする。

【0189】コントローラ330は、通常符号化処理では、オペレータによって指定されたターゲットビットレートと送信バッファ残量に基づいて、量子化回路317の量子化ステップサイズをコントロールしていたが、このパラメータ再利用符号化処理では、ターゲットビットレート、送信バッファ残量及びヒストリ情報に含まれている過去の量子化スケールに基づいて、量子化回路317の量子化ステップサイズをコントロールする。なお、以下の説明において、ヒストリ情報に含まれている過去の量子化スケールをhistory\_q\_scale\_codeと記述することにする。また、後述するヒストリストリームにおいては、この量子化スケールを、quantiser\_scale\_codeと記述している。

【0190】まず、コントローラ330は、通常符号化処理と同じように、現在の量子化スケールfeedback\_q\_scale\_codeを生成する。このfeedback\_q\_scale\_codeは、この送信バッファ319がオーバーフロー及びアンダーフローしないように、この送信バッファ319のバッファ残量に応じて決定される値である。続いて、第1世代のヒストリストリームに含まれている過去の量子化スケールhistory\_q\_scale\_codeの値と、この現在の量子化スケールfeedback\_q\_scale\_codeの値を比較し、どちらの量子化スケールの方が大きいかを判断する。量子化スケールが大きいとは、量子化ステップが大きいことを意味する。もし、現在の量子化スケールfeedback\_q\_scale\_codeが、過去の量子化スケールhistory\_q\_scale\_codeよりも大きいのであれば、コントローラ330は、この現在の量子化スケールfeedback\_q\_scale\_codeを量子化回路317に供給する。一方、過去の量子化スケールhistory\_q\_scale\_codeが、現在の量子化スケールfeedback\_q\_scale\_codeよりも大きいのであれば、コントローラ330は、この過去の量子化スケールhistory\_q\_scale\_codeを量子化回路317に供給する。

【0191】つまり、コントローラ330は、ヒストリ情報に含まれている複数の過去の量子化スケールと、送信バッファの残量から計算された現在の量子化スケールの中で、もっとも大きい量子化スケールコードを選択する。また、別の言葉で説明するのであれば、コントローラ330は、過去（第1、第2及び第3世代）の符号化処理における量子化ステップまたは現在（第4世代）の符号化処理において使用された量子化ステップの中で、もっとも大きい量子化ステップを使用して量子化を行うように量子化回路317を制御する。この理由を以下に説明する。

【0192】たとえば、第3世代の符号化処理において生成されたストリームのビットレートが4 [Mbps]であって、この第4世代の符号化処理を行うエンコーダ301に対して設定されたターゲットビットレートが15 [Mbps]であったとする。このときに、ターゲットビットレートが上がっているのに、単純に量子化ステップを小さくすれば良いかというと、実際にはそうではない。過去の符号化処理において大きい量子化ステップで符号化処理されたピクチャを、現在の符号化処理において、量子化ステップを小さくして符号化処理を行ったとしても、このピクチャの画質は向上することはない。つまり、過去の符号化処理における量子化ステップよりも小さい量子化ステップで符号化することは、単にビット量が増えるだけであって、画質を向上させることにはならない。よって、過去（第1、第2及び第3世代）の符号化処理における量子化ステップまたは現在（第4世代）の符号化処理において使用された量子化ステップの中で、もっとも大きい量子化ステップを使用して量子化を行うと、もっとも効率の良い符号化処理が行える。

【0193】更に、エンコーダ301は、「パラメータ再利用符号化処理」を実行している場合、比較装置114から供給される不連続パラメータを基に、不連続な画像の前後で「通常符号化処理」を実行するので、不連続な点の前後の画像の劣化を防止することができる。

【0194】次に、図4におけるヒストリデコーディング装置104とヒストリエンコーディング装置117についてさらに説明する。同図に示すように、ヒストリデコーディング装置104は、復号装置102より供給されるユーザデータをデコードするユーザデータデコーダ201、ユーザデータデコーダ201の出力を変換するコンバータ202、およびコンバータ202の出力から履歴情報を再生するヒストリVLD203により構成されている。

【0195】また、ヒストリエンコーディング装置117は、ヒストリ情報分離装置115より供給される3世代分の符号化パラメータをフォーマット化するヒストリVLC211、ヒストリVLC211の出力を変換するコンバータ212、コンバータ212の出力をユーザデータのフォーマットにフォーマットするユーザデータフォーマッタ213により構成されている。

【0196】ユーザデータデコーダ201は、復号装置102より供給されるユーザデータをデコードして、コンバータ202に出力する。詳細は図31を参照して後述するが、ユーザデータ（user\_data()）は、user\_data\_start\_codeとuser\_dataからなり、MPEG規格においてはuser\_dataの中に、連続する23ビットの”0”（start\_codeと同一のコード）を発生させることを禁止している。これは、そのデータが、start\_codeとして誤検出されるのを防止するためである。履歴情報（history\_stream()）は、ユーザデータエリアに（MPEG規格のuser\_dat

aの一種として）記述され、その中には、このような連続する23ビット以上の”0”が存在することがあり得るので、これを、連続する23ビット以上の”0”が発生しないように、所定のタイミングで”1”を挿入処理して、converted\_history\_stream()（後述する図18）に変換する必要がある。この変換を行うのは、ヒストリエンコーディング装置117のコンバータ212である。ヒストリデコーディング装置104のコンバータ202は、このコンバータ212と逆の変換処理を行う（連続する23ビット以上の”0”を発生させないために挿入された”1”を除去する）ものである。

【0197】ヒストリVLD203は、コンバータ202の出力から履歴情報（いまの場合、第1世代の符号化パラメータと第2世代の符号化パラメータ）を生成し、ヒストリ情報多重化装置103に出力する。

【0198】一方、ヒストリエンコーディング装置117においては、ヒストリVLC211がヒストリ情報分離装置115より供給される3世代分の（第1世代、第2世代、および第3世代の）符号化パラメータを履歴情報のフォーマットに変換する。このフォーマットには、固定長のもの（後述する図20乃至図26）と、可変長のもの（後述する図27）とがある。これらの詳細については後述する。

【0199】ヒストリVLC211により、フォーマット化された履歴情報は、コンバータ212において、converted\_history\_stream()に変換される。これは、上述したように、user\_data()のstart\_codeが誤検出されないようにするための処理である。すなわち、履歴情報内には連続する23ビット以上の”0”が存在するが、user\_data中には連続する23ビット以上の”0”を配置することができないので、この禁止項目に触れないようにコンバータ212によりデータを変換する（”1”を所定のタイミングで挿入する）のである。

【0200】ユーザデータフォーマッタ213は、コンバータ212より供給されるconverted\_history\_stream()に、後述する図18に基づいて、History\_Data\_IDを付加し、さらに、user\_data\_stream\_codeを付加して、video\_stream中に挿入できるMPEG規格のuser\_dataを生成し、符号化装置116に出力する。

【0201】図15は、例えば映像編集スタジオにおいて、複数のトランスコーディングシステム1-1乃至1-Nが直列に接続されて使用される状態を示している。各トランスコーディングシステム1-i（i=1乃至N）のヒストリ情報多重化装置103-iは、上述した符号化パラメータ用の領域の最も古い符号化パラメータが記録されている区画に、自己が用いた最新の符号化パラメータを上書きする。このことにより、ベースバンドの画像データには、同一のマクロブロックに対応する直近の4世代分の符号化パラメータ（世代履歴情報）が記録されることになる（図7）。

【0202】各符号化装置116-iのエンコーダ301-i(図14)は、その可変長符号化回路318において、ヒストリ情報分離装置115-iから供給される今回用いる符号化パラメータに基づいて、量子化回路317より供給されるビデオデータを符号化する。このようにして生成されるビットストリーム(例えば、picture\_header())中に、その現符号化パラメータは多重化される。

【0203】可変長符号化回路318はまた、ヒストリエンコーディング装置117-iより供給されるユーザデータ(世代履歴情報を含む)を、出力するビットストリーム中に多重化する(図7に示すような埋め込み処理ではなく、ビットストリーム中に多重化する)。そして、符号化装置116-iの出力するビットストリームは、SDTI(Serial Data Transfer Interface)351-iを介して、後段のトランスコーディングシステム1-(i+1)に入力される。

【0204】トランスコーディングシステム1-iとトランスコーディングシステム1-(i+1)は、それぞれ図4に示すように構成されている。従って、その処理は、図4を参照して説明した場合と同様となる。

【0205】実際の符号化パラメータの履歴を利用した符号化として、現在Iピクチャとして符号化されていたものを、PもしくはBピクチャに変更したい場合、過去の符号化パラメータの履歴を見て、過去にPもしくはBピクチャであった場合を探し、これらの履歴が存在した場合は、その動きベクトルなどのパラメータを利用して、ピクチャタイプを変更する。反対に過去に履歴がない場合は、動き検出を行わないピクチャタイプの変更を断念する。もちろん履歴がない場合であっても、動き検出を行えばピクチャタイプを変更できる。

【0206】なお、本実施の形態におけるトランスコーディングシステム1の内部においては、上述したように、復号側と符号側が粗結合されており、符号化パラメータを画像データに多重化させて伝送させたが、図16に示すように、復号装置102と符号化装置116を直接接続する(密結合する)ようにしてもよい。

【0207】図4において説明したトランスコーディングシステム1は、第1世代から第3世代の過去の符号化パラメータを符号化装置116に供給するために、ベースバンドビデオデータに過去の符号化パラメータを多重化して伝送するようにしていた。しかしながら、本発明においては、ベースバンドビデオデータに過去の符号化パラメータを多重化する技術は必須ではなく、図16に示されたように、ベースバンドビデオデータとは異なる伝送路(たとえばデータ転送バス)を使用して、過去の符号化パラメータを伝送するようにしても良い。

【0208】つまり、図16に示した、復号装置102、ヒストリデコーディング装置104、符号化装置116及びヒストリエンコーディング装置117は、図4

において説明した復号装置102、ヒストリデコーディング装置104、符号化装置116及びヒストリエンコーディング装置117とまったく同じ機能及び構成を有している。

【0209】復号装置102の可変長復号回路262は、第3世代の符号化ストリームST(3rd)のシーケンス層、GOP層、ピクチャ層、スライス層及びマクロブロック層から、第3世代の符号化パラメータを抽出し、それを、ヒストリエンコーディング装置117及び符号化装置116のコントローラ330にそれぞれ供給する。

【0210】復号装置102の可変長復号回路262は、第3世代の符号化ストリームSTから分離されたカウンタ値をフォーマット変換装置361に供給するとともに、フレームまたはフィールドに同期したFrame/Field同期信号をカウンタ362に供給する。

【0211】カウンタ362は、可変長復号回路262から供給されるフレームまたはフィールドに同期したFrame/Field同期信号を基に、カウントアップされる16ビットのカウンタである。カウンタ362は、0乃至65,535のいずれかのカウンタ値を比較装置363に出力する。

【0212】カウンタ362は、65,535のカウンタ値を有する場合、可変長復号回路262からフレームまたはフィールドに同期したFrame/Field同期信号が供給されたとき、カウンタ値を0とし、その後も、可変長復号回路262から供給されるFrame/Field同期信号を基に、カウントアップを継続する。

【0213】なお、カウンタ362は、可変長復号回路262から供給されるフレームまたはフィールドに同期したFrame/Field同期信号を基に、カウントダウン(1を減算)するようにしてもよい。

【0214】図17は、カウンタ362の構成例を示す図である。カウンタ381は、Clock信号によりカウントアップされる、16ビットのバイナリカウンタである。カウンタ381の全ての出力が"1"になったとき(すなわち、出力が65,535となったとき)、AND回路382の出力が"1"になるので、カウンタ381は、リセットされる(すなわち、カウンタ値が0になる)。

【0215】なお、カウンタ101、カウンタ113、および後述するカウンタ364も、カウンタ362と同様に構成することかできる。

【0216】フォーマット変換装置361は、可変長復号回路262から供給された、第3世代の符号化ストリームSTから分離されたカウンタ値の方式を16ビットのカウンタ値(例えば、カウンタ362が出力するカウンタ値と同じ方式)に変換して、比較装置363に出力する。

【0217】比較装置363は、フォーマット変換装置361から供給されたカウンタ値とカウンタ362から供給されたカウンタ値とを比較し、その値が同一である

とき、所定の値の不連続パラメータを符号化装置 1 1 6 に出力し、その値が異なるとき、他の値の不連続パラメータを符号化装置 1 1 6 に出力する。

【0 2 1 8】比較装置 3 6 3 にフォーマット変換装置 3 6 1 から供給されたカウンタ値とカウンタ 3 6 2 から供給されたカウンタ値とが異なる値をとるとき、カウンタ 3 6 2 は、フォーマット変換装置 3 6 1 が出力するカウンタ値をロードして、自己のカウンタ値として設定する。このようにすることで、比較装置 3 6 3 は、一旦、画像の不連続点を検出した後でも、再度、画像に不連続点が含まれていた場合、他の値の不連続パラメータを符号化装置 1 1 6 に出力することができる。

【0 2 1 9】ヒストリエンコーディング装置 1 1 7 は、受け取った第 3 世代の符号化パラメータをピクチャ層のユーザデータエリアに記述できるように converted\_history\_stream() に変換し、converted\_history\_stream() をユーザデータとして符号化装置 1 1 6 の可変長符号化回路 3 1 8 に供給する。

【0 2 2 0】さらに可変長復号回路 2 6 2 は、第 3 世代の符号化ストリームのピクチャ層のユーザデータエリアから、第 1 世代の符号化パラメータ及び第 2 の符号化パラメータを含んでいるユーザデータ user\_data を抽出し、ヒストリデコーディング装置 1 0 4 及び符号化装置 1 1 6 の可変長符号化回路 3 1 8 に供給する。ヒストリデコーディング装置 1 0 4 は、ユーザデータエリアに converted\_history\_stream() として記述されたヒストリストリームから、第 1 世代の符号化パラメータ及び第 2 世代の符号化パラメータを抽出し、それを符号化装置 1 1 6 のコントローラに供給する。

【0 2 2 1】符号化装置 1 1 6 のコントローラ 3 3 0 は、ヒストリデコーディング装置 1 0 4 から受け取った第 1 世代及び第 2 世代の符号化パラメータと、符号化装置 1 0 2 から受け取った第 3 世代の符号化パラメータとに基づいて、符号化装置 1 1 6 の符号化処理をコントロールする。

【0 2 2 2】符号化装置 1 1 6 の可変長符号化回路 3 1 8 は、復号装置 1 0 2 から第 1 世代の符号化パラメータ及び第 2 の符号化パラメータを含んでいるユーザデータ user\_data を受け取るとともに、ヒストリエンコーディング装置 1 1 7 から第 3 世代の符号化パラメータを含んでいるユーザデータ user\_data を受け取り、それらのユーザデータをヒストリ情報として、第 4 世代の符号化ストリームのピクチャ層のユーザデータエリアに記述する。

【0 2 2 3】また、符号化装置 1 1 6 は、カウンタ 3 6 4 にフレームまたはフィールドに同期した Frame/Field 同期信号を供給する。

【0 2 2 4】カウンタ 3 6 4 は、符号化装置 1 1 6 から供給されるフレームまたはフィールドに同期した Frame/Field 同期信号を基に、カウントアップされる 1 6 ビッ

トのカウンタである。カウンタ 3 6 4 は、0 乃至 65, 535 のいずれかのカウンタ値をフォーマット変換装置 3 6 5 に出力する。

【0 2 2 5】カウンタ 3 6 4 は、65, 535 のカウンタ値を有する場合、符号化装置 1 1 6 からフレームまたはフィールドに同期した Frame/Field 同期信号が供給されたとき、カウンタ値を 0 とし、その後も、符号化装置 1 1 6 から供給される Frame/Field 同期信号を基に、カウントアップを継続する。

【0 2 2 6】なお、カウンタ 3 6 4 は、符号化装置 1 1 6 から供給されるフレームまたはフィールドに同期した Frame/Field 同期信号を基に、カウントダウン (1 を減算) するようにしてもよい。

【0 2 2 7】フォーマット変換装置 3 6 5 は、カウンタ 3 6 4 から供給された 1 6 ビットのカウンタ値を、符号化ストリーム S T に多重化できる方式に変換して、符号化装置 1 1 6 に出力する。

【0 2 2 8】符号化装置 1 1 6 は、フォーマット変換装置 3 6 5 から供給されたカウンタ値を第 4 世代の符号化ストリームに格納する。

【0 2 2 9】図 1 8 は、MPEG のビデオストリームをデコードするためのシンタックスを表わした図である。デコードは、このシンタックスに従って MPEG ビットストリームをデコードすることによって、ビットストリームから意味のある複数のデータ項目 (データエレメント) を抽出する。以下に説明するシンタックスは、図において、その関数や条件文は細活字で表わされ、そのデータエレメントは、太活字で表されている。データ項目は、その名称、ビット長、及びそのタイプと伝送順序を示すユニモニック (Mnemonic) で記述されている。

【0 2 3 0】まず、この図 1 8 に示されているシンタックスにおいて使用されている関数について説明する。

【0 2 3 1】next\_start\_code() 関数は、ビットストリーム中に記述されているスタートコードを探すための関数である。この図 1 8 に示されたシンタックスにおいて、この next\_start\_code() 関数の次に、sequence\_header() 関数と sequence\_extension() 関数とが順に配置されているので、このビットストリームには、この sequence\_header() 関数と sequence\_extension() 関数によって定義されたデータエレメントが記述されている。従って、ビットストリームのデコード時には、この next\_start\_code() 関数によって、sequence\_header() 関数と sequence\_extension() 関数の先頭に記述されているスタートコード (データエレメントの一種) をビットストリーム中から見つけ、それを基準にして、sequence\_header() 関数と sequence\_extension() 関数をさらに見つけ、それらによって定義された各データエレメントをデコードする。

【0 2 3 2】尚、sequence\_header() 関数は、MPEG ビットストリームのシーケンス層のヘッダデータを定義するための関数であって、sequence\_extension() 関数は、MP

EGビットストリームのシーケンス層の拡張データを定義するための関数である。

【0 2 3 3】sequence\_extension()関数の次に配置されている do{}while構文は、while文によって定義されている条件が真である間、do文の{}内の関数に基いて記述されたデータエレメントをデータストリーム中から抽出するための構文である。すなわち、do{}while構文によって、while文によって定義されている条件が真である間、ビットストリーム中から、do文内の関数に基いて記述されたデータエレメントを抽出するデコード処理が行われる。

【0 2 3 4】このwhile文に使用されているnextbits()関数は、ビットストリーム中に現れるビット又はビット列と、次にデコードされるデータエレメントとを比較するための関数である。この図 1 8 のシンタックスの例では、nextbits()関数は、ビットストリーム中のビット列とビデオシーケンスの終わりを示すsequence\_end\_codeとを比較し、ビットストリーム中のビット列とsequence\_end\_codeとが一致しないときに、このwhile文の条件が真となる。従って、sequence\_extension()関数の次に配置されている do{}while構文は、ビットストリーム中に、ビデオシーケンスの終わりを示すsequence\_end\_codeが現れない間、do文中の関数によって定義されたデータエレメントがビットストリーム中に記述されていることを示している。

【0 2 3 5】ビットストリーム中には、sequence\_extension()関数によって定義された各データエレメントの次には、extension\_and\_user\_data(0)関数によって定義されたデータエレメントが記述されている。このextension\_and\_user\_data(0)関数は、MPEGビットストリームのシーケンス層の拡張データとユーザデータを定義するための関数である。

【0 2 3 6】このextension\_and\_user\_data(0)関数の次に配置されている do{}while構文は、while文によって定義されている条件が真である間、do文の{}内の関数に基いて記述されたデータエレメントを、ビットストリーム中から抽出するための関数である。このwhile文において使用されているnextbits()関数は、ビットストリーム中に現れるビット又はビット列と、picture\_start\_code又はgroup\_start\_codeとの一致を判断するための関数であって、ビットストリーム中に現れるビット又はビット列と、picture\_start\_code又はgroup\_start\_codeとが一致する場合には、while文によって定義された条件が真となる。よって、このdo{}while構文は、ビットストリーム中において、picture\_start\_code又はgroup\_start\_codeが現れた場合には、そのスタートコードの次に、do文中の関数によって定義されたデータエレメントのコードが記述されているので、このpicture\_start\_code又はgroup\_start\_codeによって示されるスタートコードを探し出すことによって、ビットストリーム中からdo

文中に定義されたデータエレメントを抽出することができる。

【0 2 3 7】このdo文の最初に記述されているif文は、ビットストリーム中にgroup\_start\_codeが現れた場合、という条件を示している。このif文による条件が真である場合には、ビットストリーム中には、このgroup\_start\_codeの次にgroup\_of\_picture\_header(1)関数及びextension\_and\_user\_data(1)関数によって定義されているデータエレメントが順に記述されている。

10 【0 2 3 8】このgroup\_of\_picture\_header(1)関数は、MPEGビットストリームのGOP層のヘッダデータを定義するための関数であって、extension\_and\_user\_data(1)関数は、MPEGビットストリームのGOP層の拡張データ(extension\_data)及びユーザデータ(user\_data)を定義するための関数である。

20 【0 2 3 9】さらに、このビットストリーム中には、group\_of\_picture\_header(1)関数及びextension\_and\_user\_data(1)関数によって定義されているデータエレメントの次に、picture\_header()関数とpicture\_coding\_extension()関数によって定義されたデータエレメントが記述されている。もちろん、先に説明したif文の条件が真とならない場合には、group\_of\_picture\_header(1)関数及びextension\_and\_user\_data(1)関数によって定義されているデータエレメントは記述されていないので、extension\_and\_user\_data(0)関数によって定義されているデータエレメントの次に、picture\_header()関数とpicture\_coding\_extension()関数によって定義されたデータエレメントが記述されている。

30 【0 2 4 0】このpicture\_header()関数は、MPEGビットストリームのピクチャ層のヘッダデータを定義するための関数であって、picture\_coding\_extension()関数は、MPEGビットストリームのピクチャ層の第1の拡張データを定義するための関数である。

40 【0 2 4 1】次のwhile文は、このwhile文によって定義されている条件が真である間、次のif文の条件判断を行うための関数である。このwhile文において使用されているnextbits()関数は、ビットストリーム中に現れるビット列と、extension\_start\_code又はuser\_data\_start\_codeとの一致を判断するための関数であって、ビットストリーム中に現れるビット列と、extension\_start\_code又はuser\_data\_start\_codeとが一致する場合には、このwhile文によって定義された条件が真となる。

【0 2 4 2】第1のif文は、ビットストリーム中に現れるビット列とextension\_start\_codeとの一致を判断するための関数である。ビットストリーム中に現れるビット列と32ビットのextension\_start\_codeとが一致する場合には、ビットストリーム中において、extension\_start\_codeの次にextension\_data(2)関数によって定義されるデータエレメントが記述されている。

50 【0 2 4 3】第2のif文は、ビットストリーム中に現れ

るビット列とuser\_data\_start\_codeとの一致を判断するための構文であって、ビットストリーム中に現れるビット列と32ビットのuser\_data\_start\_codeとが一致する場合には、第3のif文の条件判断が行われる。このuser\_data\_start\_codeは、MPEGビットストリームのピクチャ層のユーザデータエリアの開始を示すためのスタートコードである。

【0244】video\_continuity\_counter\_IDは、user\_data()がvideo\_continuity\_counter()であることを識別するために利用される。marker\_bit\_1は、“1”が設定され、スタートコードのエミュレーションを防止する。video\_continuity\_counterには、カウンタ値が設定される。marker\_bit\_2は、“1”が設定され、スタートコードのエミュレーションを防止する。

【0245】video\_continuity\_counterには、video\_continuity\_counter\_ID乃至marker\_bit\_2に対応したエラーをチェックするための、CRC(Cyclic Redundancy Check)が設定される。

【0246】第3のif文は、ビットストリーム中に現れるビット列とHistory\_Data\_IDとの一致を判断するための構文である。ビットストリーム中に現れるビット列とこの32ビットのHistory\_Data\_IDとが一致する場合には、このMPEGビットストリームのピクチャ層のユーザデータエリアにおいて、この32ビットのHistory\_Data\_IDによって示されるコードの次に、converted\_history\_stream()関数によって定義されるデータエレメントが記述されている。

【0247】converted\_history\_stream()関数は、MPEG符号化時に使用したあらゆる符号化パラメータを伝送するための履歴情報及び履歴データを記述するための関数である。このconverted\_history\_stream()関数によって定義されているデータエレメントの詳細は、図20乃至図27を参照して、history\_stream()として後述する。また、このHistory\_Data\_IDは、MPEGビットストリームのピクチャ層のユーザデータエリアに記述されたこの履歴情報及び履歴データが記述されている先頭を示すためのスタートコードである。

【0248】else文は、第3のif文において、条件が非真であることを示すための構文である。従って、このMPEGビットストリームのピクチャ層のユーザデータエリアにおいて、converted\_history\_stream()関数によって定義されたデータエレメントが記述されていない場合には、user\_data()関数によって定義されたデータエレメントが記述されている。

【0249】図18において、履歴情報は、converted\_history\_stream()に記述され、user\_data()に記述される訳ではないが、このconverted\_history\_stream()は、MPEG規格のuser\_dataの一種として記述される。そこで、本明細書中においては、場合によって、履歴情報がuser\_dataに記述されるとも説明するが、それは、MPEG

規格のuser\_dataの一種として記述されるということの意味する。

【0250】picture\_data()関数は、MPEGビットストリームのピクチャ層のユーザデータの次に、スライス層及びマクロブロック層に関するデータエレメントを記述するための関数である。通常は、このpicture\_data()関数によって示されるデータエレメントは、ビットストリームのピクチャ層のユーザデータエリアに記述されたconverted\_history\_stream()関数によって定義されるデータエレメント又はuser\_data()関数によって定義されたデータエレメントの次に記述されているが、ピクチャ層のデータエレメントを示すビットストリーム中に、extension\_start\_code又はuser\_data\_start\_codeが存在しない場合には、このpicture\_data()関数によって示されるデータエレメントは、picture\_coding\_extension()関数によって定義されるデータエレメントの次に記述されている。

【0251】このpicture\_data()関数によって示されるデータエレメントの次には、sequence\_header()関数とsequence\_extension()関数とによって定義されたデータエレメントが順に配置されている。このsequence\_header()関数とsequence\_extension()関数によって記述されたデータエレメントは、ビデオストリームのシーケンスの先頭に記述されたsequence\_header()関数とsequence\_extension()関数によって記述されたデータエレメントと全く同じである。このように同じデータをストリーム中に記述する理由は、ビットストリーム受信装置側でデータストリームの途中(例えばピクチャ層に対応するビットストリーム部分)から受信が開始された場合に、シーケンス層のデータを受信できなくなり、ストリームをデコード出来なくなることを防止するためである。

【0252】この最後のsequence\_header()関数とsequence\_extension()関数とによって定義されたデータエレメントの次、つまり、データストリームの最後には、シーケンスの終わりを示す32ビットのsequence\_end\_codeが記述されている。

【0253】以上のシンタックスの基本的な構成の概略を示すと、図19に示すようになる。

【0254】次に、converted\_history\_stream()関数によって定義されたヒストリストリームに関して説明する。

【0255】このconverted\_history\_stream()は、MPEGのピクチャ層のユーザデータエリアに履歴情報を示すヒストリストリームを挿入するための関数である。尚、「converted」の意味は、スタートエミュレーションを防止するために、ユーザエリアに挿入すべき履歴データから構成される履歴ストリームの少なくとも22ビット毎にマーカービット(1ビット)を挿入する変換処理を行ったストリームであることを意味している。

【0256】このconverted\_history\_stream()は、以下

に説明する固定長の履歴ストリーム（図 2 0 乃至図 2 6）又は可変長の履歴ストリーム（図 2 7）のいずれかの形式で記述される。エンコーダ側において固定長の履歴ストリームを選択した場合には、デコーダ側において履歴ストリームから各データエレメントをデコードするための回路及びソフトウェアが簡単になるというメリットがある。一方、エンコーダ側において可変長の履歴ストリームを選択した場合には、エンコーダにおいてピクチャ層のユーザエリアに記述される履歴情報（データエレメント）を必要に応じて任意に選択することができるので、履歴ストリームのデータ量を少なくすることができ、その結果、符号化されたビットストリーム全体のデータレートを低減することができる。

【0 2 5 7】本発明において説明する「履歴ストリーム」、「ヒストリストリーム」、「履歴情報」、「ヒストリ情報」、「履歴データ」、「ヒストリデータ」、「履歴パラメータ」、「ヒストリパラメータ」とは、過去の符号化処理において使用した符号化パラメータ（又はデータエレメント）を意味し、現在の（最終段の）符号化処理において使用した符号化パラメータを意味するものではない。例えば、第 1 世代の符号化処理において、あるピクチャを 1 ピクチャで符号化して伝送し、次なる第 2 世代の符号化処理において、このピクチャを今度は P ピクチャとして符号化して伝送し、さらに、第 3 世代の符号化処理において、このピクチャを B ピクチャで符号化して伝送する例をあげて説明する。

【0 2 5 8】第 3 世代の符号化処理において使用した符号化パラメータが、第 3 世代の符号化処理において生成された符号化ビットストリームのシーケンス層、GOP 層、ピクチャ層、スライス層及びマクロブロック層の所定位置に記述されている。一方、過去の符号化処理である第 1 世代及び第 2 世代の符号化処理において使用した符号化パラメータが記述されるシーケンス層や GOP 層に記述されるのでは無く、既に説明したシンタックスに従って、符号化パラメータの履歴情報として、ピクチャ層のユーザデータエリアに記述される。

【0 2 5 9】まず、固定長の履歴ストリームシンタックスについて図 2 0 乃至図 2 6 を参照して説明する。

【0 2 6 0】最終段（例えば第 3 世代）の符号化処理において生成されたビットストリームのピクチャ層のユーザデータエリアには、まず最初に、過去（例えば第 1 世代及び第 2 世代）の符号化処理において使用されていたシーケンス層のシーケンスヘッダに含められる符号化パラメータが、履歴ストリームとして挿入される。尚、過去の符号化処理において生成されたビットストリームのシーケンス層のシーケンスヘッダ等の履歴情報は、最終段の符号化処理において生成されたビットストリームのシーケンス層のシーケンスヘッダに挿入されることは無いという点に注意すべきである。

【0 2 6 1】過去の符号化処理で使用したシーケンスヘッダ（sequence\_header）に含められるデータエレメントは、sequence\_header\_code、sequence\_header\_present\_flag、horizontal\_size\_value、marker\_bit、vertical\_size\_value、aspect\_ratio\_information、frame\_rate\_code、bit\_rate\_value、VBV\_buffer\_size\_value、constrained\_parameter\_flag、load\_intra\_quantiser\_matrix、load\_non\_intra\_quantiser\_matrix、intra\_quantiser\_matrix、及びnon\_intra\_quantiser\_matrix等から構成される。

【0 2 6 2】sequence\_header\_codeは、シーケンス層のスタート同期コードを表すデータである。sequence\_header\_present\_flagは、sequence\_header内のデータが有効か無効かを示すデータである。horizontal\_size\_valueは、画像の水平方向の画素数の下位 12 ビットから成るデータである。marker\_bitは、スタートコードエミュレーションを防止するために挿入されるビットデータである。vertical\_size\_valueは、画像の縦のライン数の下位 12 ビットからなるデータである。aspect\_ratio\_informationは、画素のアスペクト比（縦横比）または表示画面アスペクト比を表すデータである。frame\_rate\_codeは、画像の表示周期を表すデータである。

【0 2 6 3】bit\_rate\_valueは、発生ビット量に対する制限のためのビット・レートの下位 18 ビット（400 bps 単位で切り上げる）データである。VBV\_buffer\_size\_valueは、発生符号量制御用の仮想バッファ（ビデオバッファベリファイヤー）の大きさを決める値の下位 10 ビットデータである。constrained\_parameter\_flagは、各パラメータが制限以内であることを示すデータである。load\_intra\_quantiser\_matrixは、イントラ MB 用量子化マトリックス・データの存在を示すデータである。load\_non\_intra\_quantiser\_matrixは、非イントラ MB 用量子化マトリックス・データの存在を示すデータである。intra\_quantiser\_matrixは、イントラ MB 用量子化マトリックスの値を示すデータである。non\_intra\_quantiser\_matrixは、非イントラ MB 用量子化マトリックスの値を表すデータである。

【0 2 6 4】最終段の符号化処理において生成されたビットストリームのピクチャ層のユーザデータエリアには、過去の符号化処理において使用されたシーケンス層のシーケンスエクステンションを表わすデータエレメントが、履歴ストリームとして記述される。

【0 2 6 5】この過去の符号化処理で使用したシーケンスエクステンション（sequence\_extension）を表わすデータエレメントは、extension\_start\_code、extension\_start\_code\_identifier、sequence\_extension\_present\_flag、profile\_and\_level\_indication、progressive\_sequence、chroma\_format、horizontal\_size\_extension、vertical\_size\_extension、bit\_rate\_extension、vbv\_buffer\_size\_extension、low\_delay、frame\_rate\_ext



ension\_n、及び frame\_rate\_extension\_d等のデータエレメントである。

【0 2 6 6】extension\_start\_codeは、エクステンションデータのスタート同期コードを表すデータである。extension\_start\_code\_identifierは、どの拡張データが送られるかを示すデータである。sequence\_extension\_present\_flagは、シーケンスエクステンション内のデータが有効であるか無効であるかを示すデータである。profile\_and\_level\_indicationは、ビデオデータのプロファイルとレベルを指定するためのデータである。progressive\_sequenceは、ビデオデータが順次走査であることを示すデータである。chroma\_formatは、ビデオデータの色差フォーマットを指定するためのデータである。

【0 2 6 7】horizontal\_size\_extensionは、シーケンスヘッダのhorizontal\_size\_valueに加える上位 2 ビットのデータである。vertical\_size\_extensionは、シーケンスヘッダのvertical\_size\_valueに加える上位 2 ビットのデータである。bit\_rate\_extensionは、シーケンスヘッダのbit\_rate\_valueに加える上位 1 2 ビットのデータである。vbv\_buffer\_size\_extensionは、シーケンスヘッダのvbv\_buffer\_size\_valueに加える上位 8 ビットのデータである。low\_delayは、B ピクチャを含まないことを示すデータである。frame\_rate\_extension\_nは、シーケンスヘッダのframe\_rate\_codeと組み合わせてフレームレートを取得するためのデータである。frame\_rate\_extension\_dは、シーケンスヘッダのframe\_rate\_codeと組み合わせてフレームレートを取得するためのデータである。

【0 2 6 8】続いて、ビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたシーケンス層のシーケンスディスプレイエクステンションを表わすデータエレメントが、履歴ストリームとして記述される。

【0 2 6 9】このシーケンスディスプレイエクステンション(sequence\_display\_extension)として記述されているデータエレメントは、extension\_start\_code、extension\_start\_code\_identifier、sequence\_display\_extension\_present\_flag、video\_format、colour\_description、colour\_primaries、transfer\_characteristics、matrix\_coefficients、display\_horizontal\_size、及びdisplay\_vertical\_sizeから構成される。

【0 2 7 0】extension\_start\_codeは、エクステンションデータのスタート同期コードを表すデータである。extension\_start\_code\_identifierは、どの拡張データが送られるかを示すコードである。sequence\_display\_extension\_present\_flagは、シーケンスディスプレイエクステンション内のデータエレメントが有効か無効かを示すデータである。video\_formatは、原信号の映像フォーマットを表すデータである。color\_descriptionは、色空間の詳細データがあることを示すデータである。colo

r\_primariesは、原信号の色特性の詳細を示すデータである。transfer\_characteristicsは、光電変換がどのように行われたのかの詳細を示すデータである。matrix\_coefficientsは、原信号が光の三原色からどのように変換されたかの詳細を示すデータである。display\_horizontal\_sizeは、意図するディスプレイの活性領域(水平サイズ)を表すデータである。display\_vertical\_sizeは、意図するディスプレイの活性領域(垂直サイズ)を表すデータである。

10 【0 2 7 1】続いて、最終段の符号化処理において生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において生成されたマクロブロックの位相情報を示すマクロブロックアサインメントデータ(macroblock\_assignment\_in\_user\_data)が、履歴ストリームとして記述される。

【0 2 7 2】このマクロブロックの位相情報を示すmacroblock\_assignment\_in\_user\_dataは、macroblock\_assignment\_present\_flag、v\_phase、h\_phase等のデータエレメントから構成される。

20 【0 2 7 3】このmacroblock\_assignment\_present\_flagは、macroblock\_assignment\_in\_user\_data内のデータエレメントが有効か無効かを示すデータである。v\_phaseは、画像データからマクロブロックを切り出す際の垂直方向の位相情報を示すデータである。h\_phaseは、画像データからマクロブロックを切り出す際の水平方向の位相情報を示すデータである。

【0 2 7 4】続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたGOP層のGOPヘッダを表わすデータエレメントが、履歴ストリームとして記述されている。

【0 2 7 5】このGOPヘッダ(group\_of\_picture\_header)を表わすデータエレメントは、group\_start\_code、group\_of\_picture\_header\_present\_flag、time\_code、closed\_gop、及びbroken\_linkから構成される。

【0 2 7 6】group\_start\_codeは、GOP層の開始同期コードを示すデータである。group\_of\_picture\_header\_present\_flagは、group\_of\_picture\_header内のデータエレメントが有効であるか無効であるかを示すデータである。time\_codeは、GOPの先頭ピクチャのシーケンスの先頭からの時間を示すタイムコードである。closed\_gopは、GOP内の画像が他のGOPから独立再生可能なことを示すフラグデータである。broken\_linkは、編集などのためにGOP内の先頭のBピクチャが正確に再生できないことを示すフラグデータである。

【0 2 7 7】続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたピクチャ層のピクチャヘッダを表わすデータエレメントが、履歴ストリームとして記述されている。

【0 2 7 8】このピクチャヘッダ (picture\_header) に関するデータエレメントは、picture\_start\_code、temporal\_reference、picture\_coding\_type、vbm\_delay、full\_pel\_forward\_vector、forward\_f\_code、full\_pel\_backward\_vector、及びbackward\_f\_codeから構成される。

【0 2 7 9】具体的には、picture\_start\_codeは、ピクチャ層の開始同期コードを表すデータである。temporal\_referenceは、ピクチャの表示順を示す番号でGOPの先頭でリセットされるデータである。picture\_coding\_typeは、ピクチャタイプを示すデータである。vbm\_delayは、ランダムアクセス時の仮想バッファの初期状態を示すデータである。full\_pel\_forward\_vectorは、順方向動きベクトルの精度が整数単位か半画素単位かを示すデータである。forward\_f\_codeは、順方向動きベクトル探索範囲を表すデータである。full\_pel\_backward\_vectorは、逆方向動きベクトルの精度が整数単位か半画素単位かを示すデータである。backward\_f\_codeは、逆方向動きベクトル探索範囲を表すデータである。

【0 2 8 0】続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたピクチャ層のピクチャコーディングエクステンションが、履歴ストリームとして記述されている。

【0 2 8 1】このピクチャコーディングエクステンション (picture\_coding\_extension) に関するデータエレメントは、extension\_start\_code、extension\_start\_code\_identifier、f\_code[0][0]、f\_code[0][1]、f\_code[1][0]、f\_code[1][1]、intra\_dc\_precision、picture\_structure、top\_field\_first、frame\_predictive\_frame\_dct、concealment\_motion\_vectors、q\_scale\_type、intra\_vlc\_format、alternate\_scan、repeat\_firt\_field、chroma\_420\_type、progressive\_frame、composite\_display\_flag、v\_axis、field\_sequence、sub\_carrier、burst\_amplitude、及びsub\_carrier\_phaseから構成される。

【0 2 8 2】extension\_start\_codeは、ピクチャ層のエクステンションデータのスタートを示す開始コードである。extension\_start\_code\_identifierは、どの拡張データが送られるかを示すコードである。f\_code[0][0]は、フォワード方向の水平動きベクトル探索範囲を表すデータである。f\_code[0][1]は、フォワード方向の垂直動きベクトル探索範囲を表すデータである。f\_code[1][0]は、バックワード方向の水平動きベクトル探索範囲を表すデータである。f\_code[1][1]は、バックワード方向の垂直動きベクトル探索範囲を表すデータである。

【0 2 8 3】intra\_dc\_precisionは、DC係数の精度を表すデータである。picture\_structureは、フレームストラクチャかフィールドストラクチャかを示すデータである。フィールドストラクチャの場合は、上位フィールドか下位フィールドかもあわせて示すデータである。top\_field\_firstは、フレームストラクチャの場合、最初の

フィールドが上位か下位かを示すデータである。frame\_predictive\_frame\_dctは、フレーム・ストラクチャの場合、フレーム・モードDCTの予測がフレーム・モードだけであることを示すデータである。concealment\_motion\_vectorsは、イントラマクロブロックに伝送エラーを隠蔽するための動きベクトルがついていることを示すデータである。

【0 2 8 4】q\_scale\_typeは、線形量子化スケールを利用するか、非線形量子化スケールを利用するかを示すデータである。intra\_vlc\_formatは、イントラマクロブロックに、別の2次元VLCを使うかどうかを示すデータである。alternate\_scanは、ジグザグスキャンを使うか、オルタネート・スキャンを使うかの選択を表すデータである。repeat\_firt\_fieldは、2:3プルダウンの際に使われるデータである。chroma\_420\_typeは、信号フォーマットが4:2:0の場合、次のprogressive\_frameと同じ値、そうでない場合は0を表すデータである。progressive\_frameは、このピクチャが、順次走査できているかどうかを示すデータである。composite\_display\_flagは、ソース信号がコンポジット信号であったかどうかを示すデータである。

【0 2 8 5】v\_axisは、ソース信号が、PALの場合に使われるデータである。field\_sequenceは、ソース信号が、PALの場合に使われるデータである。sub\_carrierは、ソース信号が、PALの場合に使われるデータである。burst\_amplitudeは、ソース信号が、PALの場合に使われるデータである。sub\_carrier\_phaseは、ソース信号が、PALの場合に使われるデータである。

【0 2 8 6】続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用された量子化マトリックスエクステンションが、履歴ストリームとして記述されている。

【0 2 8 7】この量子化マトリックスエクステンション (quant\_matrix\_extension) に関するデータエレメントは、extension\_start\_code、extension\_start\_code\_identifier、quant\_matrix\_extension\_present\_flag、load\_intra\_quantiser\_matrix、intra\_quantiser\_matrix[64]、load\_non\_intra\_quantiser\_matrix、non\_intra\_quantiser\_matrix[64]、load\_chroma\_intra\_quantiser\_matrix、chroma\_intra\_quantiser\_matrix[64]、load\_chroma\_non\_intra\_quantiser\_matrix、及びchroma\_non\_intra\_quantiser\_matrix[64] から構成される。

【0 2 8 8】extension\_start\_codeは、この量子化マトリックスエクステンションのスタートを示す開始コードである。extension\_start\_code\_identifierは、どの拡張データが送られるかを示すコードである。quant\_matrix\_extension\_present\_flagは、この量子化マトリックスエクステンション内のデータエレメントが有効か無効かを示すためのデータである。load\_intra\_quantiser\_m

atrixは、イントラマクロブロック用の量子化マトリックスデータの存在を示すデータである。intra\_quantiser\_matrixは、イントラマクロブロック用の量子化マトリックスの値を示すデータである。

【0 2 8 9】load\_non\_intra\_quantiser\_matrixは、非イントラマクロブロック用の量子化マトリックスデータの存在を示すデータである。non\_intra\_quantiser\_matrixは、非イントラマクロブロック用の量子化マトリックスの値を表すデータである。load\_chroma\_intra\_quantiser\_matrixは、色差イントラマクロブロック用の量子化マトリックス・データの存在を示すデータである。chroma\_intra\_quantiser\_matrixは、色差イントラマクロブロック用の量子化マトリックスの値を示すデータである。load\_chroma\_non\_intra\_quantiser\_matrixは、色差非イントラマクロブロック用の量子化マトリックス・データの存在を示すデータである。chroma\_non\_intra\_quantiser\_matrixは、色差非イントラマクロブロック用の量子化マトリックスの値を示すデータである。

【0 2 9 0】続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたコピーライトエクステンションが、履歴ストリームとして記述されている。

【0 2 9 1】このコピーライトエクステンション (copyright\_extension) に関するデータエレメントは、extension\_start\_code、extension\_start\_code\_identifier、copyright\_extension\_present\_flag、copyright\_flag、copyright\_identifier、original\_or\_copy、copyright\_number\_1、copyright\_number\_2、及びcopyright\_number\_3から構成される。

【0 2 9 2】extension\_start\_codeは、コピーライトエクステンションのスタートを示す開始コードである。extension\_start\_code\_identifierのどのエクステンションデータが送られるかを示すコードである。copyright\_extension\_present\_flagは、このコピーライトエクステンション内のデータエレメントが有効か無効かを示すためのデータである。copyright\_flagは、次のコピーライトエクステンション又はシーケンスエンドまで、符号化されたビデオデータに対してコピー権が与えられているか否かを示す。

【0 2 9 3】copyright\_identifierは、ISO/IEC JTC/SC 29によって指定されたコピー権の登録機関を識別するためのデータである。original\_or\_copyは、ビットストリーム中のデータが、オリジナルデータであるかコピーデータであるかを示すデータである。copyright\_number\_1は、コピーライトナンバーのビット4 4から6 3を表わすデータである。copyright\_number\_2は、コピーライトナンバーのビット2 2から4 3を表わすデータである。copyright\_number\_3は、コピーライトナンバーのビット0から2 1を表わすデータである。

【0 2 9 4】続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたピクチャディスプレイエクステンション (picture\_display\_extension) が、履歴ストリームとして記述されている。

【0 2 9 5】このピクチャディスプレイエクステンションを表わすデータエレメントは、extension\_start\_code、extension\_start\_code\_identifier、picture\_display\_extension\_present\_flag、frame\_center\_horizontal\_offset\_1、frame\_center\_vertical\_offset\_1、frame\_center\_horizontal\_offset\_2、frame\_center\_vertical\_offset\_2、frame\_center\_horizontal\_offset\_3、及びframe\_center\_vertical\_offset\_3から構成される。

【0 2 9 6】extension\_start\_codeは、ピクチャディスプレイエクステンションのスタートを示すための開始コードである。extension\_start\_code\_identifierは、どの拡張データが送られるかを示すコードである。picture\_display\_extension\_present\_flagは、ピクチャディスプレイエクステンション内のデータエレメントが有効か無効かを示すデータである。frame\_center\_horizontal\_offsetは、表示エリアの水平方向のオフセットを示すデータであって、3つのオフセット値まで定義することができる。frame\_center\_vertical\_offsetは、表示エリアを垂直方向のオフセットを示すデータであって、3つのオフセット値まで定義することができる。

【0 2 9 7】最終段の符号化処理において生成されたビットストリームのピクチャ層のユーザエリアには、既に説明したピクチャディスプレイエクステンションを表わす履歴情報の次に、過去の符号化処理において使用されたユーザデータ (user\_data) が、履歴ストリームとして記述されている。

【0 2 9 8】このユーザデータの次には、過去の符号化処理において使用されたマクロブロック層に関する情報が、履歴ストリームとして記述されている。

【0 2 9 9】このマクロブロック層に関する情報は、macroblock\_address\_h、macroblock\_address\_v、slice\_header\_present\_flag、skipped\_macroblock\_flag等のマクロブロック (macroblock) の位置に関するデータエレメントと、macroblock\_quant、macroblock\_motion\_forward、macroblock\_motion\_backward、macroblock\_pattern、macroblock\_intra、spatial\_temporal\_weight\_code\_flag、frame\_motion\_type、及びdct\_type等のマクロブロックモード (macroblock\_modes[]) に関するデータエレメントと、quantiser\_scale\_code等の量子化ステップ制御に関するデータエレメントと、PMV[0][0][0]、PMV[0][0][1]、motion\_vertical\_field\_select[0][0]、PMV[0][1][0]、PMV[0][1][1]、motion\_vertical\_field\_select[0][1]、PMV[1][0][0]、PMV[1][0][1]、motion\_vertical\_field\_select[1][0]、PMV[1][1][0]、PMV[1][1][1]、motion\_vertical\_field\_select[1][1]等の動き補

償に関するデータエレメントと、coded\_block\_pattern等のマクロブロックパターンに関するデータエレメントと、num\_mv\_bits、num\_coef\_bits、及びnum\_other\_bits等の発生符号量に関するデータエレメントから構成されている。

【0 3 0 0】以下にマクロブロック層に関するデータエレメントについて詳細に説明する。

【0 3 0 1】macroblock\_address\_hは、現在のマクロブロックの水平方向の絶対位置を定義するためのデータである。macroblock\_address\_vは、現在のマクロブロックの垂直方向の絶対位置を定義するためのデータである。slice\_header\_present\_flagは、このマクロブロックがスライス層の先頭であり、スライスヘッダを伴うか否かを示すデータである。skipped\_macroblock\_flagは、復号処理においてこのマクロブロックをスキップするか否かを示すデータである。

【0 3 0 2】macroblock\_quantは、後述する図 4 3 と図 4 4 に示されたマクロブロックタイプ ( macroblock\_type ) から導かれるデータであって、quantiser\_scale\_codeがビットストリーム中に現れるか否かを示すデータである。macroblock\_motion\_forwardは、図 4 3 と図 4 4 に示されたマクロブロックタイプから導かれるデータであって、復号処理で使用されるデータである。macroblock\_motion\_backwardは、図 4 3 と図 4 4 に示されたマクロブロックタイプから導かれるデータであって、復号処理で使用されるデータである。macroblock\_patternは、図 4 3 と図 4 4 に示されたマクロブロックタイプから導かれるデータであって、coded\_block\_patternがビットストリーム中に現れるか否かを示すデータである。

【0 3 0 3】macroblock\_intraは、図 4 3 と図 4 4 に示されたマクロブロックタイプから導かれるデータであって、復号処理で使用されるデータである。spatial\_temporal\_weight\_code\_flagは、図 4 3 と図 4 4 に示されたマクロブロックタイプから導かれるデータであって、時間スケラビリティで下位レイヤ画像のアップサンプリング方法を示すspatial\_temporal\_weight\_codeは、ビットストリーム中に存在するか否かを示すデータである。

【0 3 0 4】frame\_motion\_typeは、フレームのマクロブロックの予測タイプを示す 2 ビットのコードである。予測ベクトルが 2 個でフィールドベースの予測タイプであれば「0 0」であって、予測ベクトルが 1 個でフィールドベースの予測タイプであれば「0 1」であって、予測ベクトルが 1 個でフレームベースの予測タイプであれば「1 0」であって、予測ベクトルが 1 個でディアルプライムの予測タイプであれば「1 1」である。field\_motion\_typeは、フィールドのマクロブロックの動き予測を示す 2 ビットのコードである。予測ベクトルが 1 個でフィールドベースの予測タイプであれば「0 1」であって、予測ベクトルが 2 個で 1 8 × 8 マクロブロックベースの予測タイプであれば「1 0」であって、予測ベクトル

ルが 1 個でディアルプライムの予測タイプであれば「1 1」である。dct\_typeは、DCTがフレームDCTモードか、フィールドDCTモードかを示すデータである。quantiser\_scale\_codeはマクロブロックの量子化ステップサイズを示すデータである。

【0 3 0 5】次に動きベクトルに関するデータエレメントについて説明する。動きベクトルは、復号時に必要な動きベクトルを減少させるために、先に符号化されたベクトルに関し差分として符号化される。動きベクトルの復号を行うために復号器は、4 個の動きベクトル予測値 (それぞれ水平及び垂直成分を伴う) を維持しなければいけない。この予測動きベクトルをPMV[r][s][v]と表わすことにしている。[r]は、マクロブロックにおける動きベクトルが第 1 のベクトルであるのか、第 2 のベクトルであるのかを示すフラグであって、マクロブロックにおけるベクトルが第 1 のベクトルである場合には

「0」となって、マクロブロックにおけるベクトルが第 2 のベクトルである場合には「1」となる。[s]は、マクロブロックにおける動きベクトルの方向が、前方向であるのか後方向であるのかを示すフラグであって、前方向動きベクトルの場合には「0」となって、後方向動きベクトルの場合には「1」となる。[v]は、マクロブロックにおけるベクトルの成分が、水平方向であるのか垂直方向であるのかを示すフラグであって、水平方向成分の場合には「0」となって、垂直方向成分の場合には「1」となる。

【0 3 0 6】従って、PMV[0][0][0]は、第 1 のベクトルの前方向の動きベクトルの水平方向成分のデータを表わし、PMV[0][0][1]は、第 1 のベクトルの前方向の動きベクトルの垂直方向成分のデータを表わし、PMV[0][1][0]は、第 1 のベクトルの後方向の動きベクトルの水平方向成分のデータを表わし、PMV[0][1][1]は、第 1 のベクトルの後方向の動きベクトルの垂直方向成分のデータを表わし、PMV[1][0][0]は、第 2 のベクトルの前方向の動きベクトルの水平方向成分のデータを表わし、PMV[1][0][1]は、第 2 のベクトルの前方向の動きベクトルの垂直方向成分のデータを表わし、PMV[1][1][0]は、第 2 のベクトルの後方向の動きベクトルの水平方向成分のデータを表わし、PMV[1][1][1]は、第 2 のベクトルの後方向の動きベクトルの垂直方向成分のデータを表わしている。

【0 3 0 7】motion\_vertical\_field\_select[r][s]は、予測の形式にいずれの参照フィールドを使用するかを示すデータである。このmotion\_vertical\_field\_select[r][s]が「0」の場合には、トップ参照フィールドを使用し、「1」の場合には、ボトム参照フィールドを使用することを示している。

【0 3 0 8】よって、motion\_vertical\_field\_select[0][0]は、第 1 のベクトルの前方向の動きベクトルを生成する際の参照フィールドを示し、motion\_vertical\_fi

eld\_select[0][1]は、第1のベクトルの後方向の動きベクトルを生成する際の参照フィールドを示し、motion\_vertical\_field\_select[1][0]は、第2のベクトルの前方向の動きベクトルを生成する際の参照フィールドを示し、motion\_vertical\_field\_select[1][1]は、第2ベクトルの後方向の動きベクトルを生成する際の参照フィールドを示している。

【0309】coded\_block\_patternは、DCT係数を格納する複数のDCTブロックのうち、どのDCTブロックに、有意係数（非0係数）があるかを示す可変長のデータである。num\_mv\_bitsは、マクロブロック中の動きベクトルの符号量を示すデータである。num\_coef\_bitsは、マクロブロック中のDCT係数の符号量を示すデータである。num\_other\_bitsは、マクロブロックの符号量で、動きベクトル及びDCT係数以外の符号量を示すデータである。

【0310】次に、可変長の履歴ストリームから各データエレメントをデコードするためのシンタックスについて、図27乃至図47を参照して説明する。

【0311】この可変長の履歴ストリームは、next\_start\_code()関数、sequence\_header()関数、sequence\_extension()関数、extension\_and\_user\_data()関数、group\_of\_picture\_header()関数、extension\_and\_user\_data(1)関数、picture\_header()関数、picture\_coding\_extension()関数、re\_coding\_stream\_info()関数、extension\_and\_user\_data(2)関数、及びpicture\_data()関数によって定義されたデータエレメントによって構成される。

【0312】next\_start\_code()関数は、ビットストリーム中に存在するスタートコードを探すための関数であるので、履歴ストリームの最も先頭には、図28に示すような、過去の符号化処理において使用されたデータエレメントであってsequence\_header()関数によって定義されたデータエレメントが記述されている。

【0313】sequence\_header()関数によって定義されたデータエレメントは、sequence\_header\_code、sequence\_header\_present\_flag、horizontal\_size\_value、vertical\_size\_value、aspect\_ratio\_information、frame\_rate\_code、bit\_rate\_value、marker\_bit、VBV\_buffer\_size\_value、constrained\_parameter\_flag、load\_intra\_quantiser\_matrix、intra\_quantiser\_matrix、load\_non\_intra\_quantiser\_matrix、及びnon\_intra\_quantiser\_matrix等である。

【0314】sequence\_header\_codeは、シーケンス層のスタート同期コードを表すデータである。sequence\_header\_present\_flagは、sequence\_header内のデータが有効か無効かを示すデータである。horizontal\_size\_valueは、画像の水平方向の画素数の下位12ビットから成るデータである。vertical\_size\_valueは、画像の縦のライン数の下位12ビットからなるデータである。aspect\_ratio\_informationは、画素のアスペクト比（縦横比）または表示画面アスペクト比を表すデータである。frame\_

rate\_codeは、画像の表示周期を表すデータである。bit\_rate\_valueは、発生ビット量に対する制限のためのビット・レートの下位18ビット（400bsp単位で切り上げる）データである。

【0315】marker\_bitは、スタートコードエミュレーションを防止するために挿入されるビットデータである。VBV\_buffer\_size\_valueは、発生符号量制御用の仮想バッファ（ビデオバッファペリファイヤー）の大きさを決める値の下位10ビットデータである。constrained\_parameter\_flagは、各パラメータが制限以内であることを示すデータである。load\_intra\_quantiser\_matrixは、イントラMB用量子化マトリックス・データの存在を示すデータである。intra\_quantiser\_matrixは、イントラMB用量子化マトリックスの値を示すデータである。load\_non\_intra\_quantiser\_matrixは、非イントラMB用量子化マトリックス・データの存在を示すデータである。non\_intra\_quantiser\_matrixは、非イントラMB用量子化マトリックスの値を表すデータである。

【0316】sequence\_header()関数によって定義されたデータエレメントの次には、図29で示すような、sequence\_extension()関数によって定義されたデータエレメントが、履歴ストリームとして記述されている。

【0317】sequence\_extension()関数によって定義されたデータエレメントとは、extension\_start\_code、extension\_start\_code\_identifier、sequence\_extension\_present\_flag、profile\_and\_level\_indication、progressive\_sequence、chroma\_format、horizontal\_size\_extension、vertical\_size\_extension、bit\_rate\_extension、vbv\_buffer\_size\_extension、low\_delay、frame\_rate\_extension\_n、及びframe\_rate\_extension\_d等のデータエレメントである。

【0318】extension\_start\_codeは、エクステンションデータのスタート同期コードを表すデータである。extension\_start\_code\_identifierは、どの拡張データが送られるかを示すデータである。sequence\_extension\_present\_flagは、シーケンスエクステンション内のデータが有効であるか無効であるかを示すデータである。profile\_and\_level\_indicationは、ビデオデータのプロファイルとレベルを指定するためのデータである。progressive\_sequenceは、ビデオデータが順次走査であることを示すデータである。chroma\_formatは、ビデオデータの色差フォーマットを指定するためのデータである。horizontal\_size\_extensionは、シーケンスヘッダのhorizontal\_size\_valueに加える上位2ビットのデータである。vertical\_size\_extensionは、シーケンスヘッダのvertical\_size\_valueに加える上位2ビットのデータである。bit\_rate\_extensionは、シーケンスヘッダのbit\_rate\_valueに加える上位12ビットのデータである。vbv\_buffer\_size\_extensionは、シーケンスヘッダのvbv\_buffer\_size\_valueに加える上位8ビットのデータである。

【0 3 1 9】low\_delayは、Bピクチャを含まないことを示すデータである。frame\_rate\_extension\_nは、シーケンスヘッダのframe\_rate\_codeと組み合わせてフレームレートを得るためのデータである。frame\_rate\_extension\_dは、シーケンスヘッダのframe\_rate\_codeと組み合わせてフレームレートを得るためのデータである。

【0 3 2 0】sequence\_extension()関数によって定義されたデータ要素の次には、図 3 0 に示すようなextension\_and\_user\_data()関数によって定義されたデータ要素が、履歴ストリームとして記述されている。extension\_and\_user\_data(i)関数は、「i」が1以外のときは、extension\_data()関数によって定義されるデータ要素は記述せずに、user\_data()関数によって定義されるデータ要素のみを履歴ストリームとして記述する。よって、extension\_and\_user\_data()関数は、user\_data()関数によって定義されるデータ要素のみを履歴ストリームとして記述する。

【0 3 2 1】user\_data()関数は、図 3 1 に示されたようなシンタックスに基いて、ユーザデータを履歴ストリームとして記述する。

【0 3 2 2】extension\_and\_user\_data()関数によって定義されたデータ要素の次には、図 3 2 に示すようなgroup\_of\_picture\_header()関数によって定義されたデータ要素、及びextension\_and\_user\_data(1)関数によって定義されるデータ要素が、履歴ストリームとして記述されている。但し、履歴ストリーム中に、GOP層のスタートコードを示すgroup\_start\_codeが記述されている場合のみ、group\_of\_picture\_header()関数によって定義されたデータ要素、及びextension\_and\_user\_data(1)関数によって定義されるデータ要素が記述されている。

【0 3 2 3】group\_of\_picture\_header()関数によって定義されたデータ要素は、group\_start\_code、group\_of\_picture\_header\_present\_flag、time\_code、closed\_gop、及びbroken\_linkから構成される。

【0 3 2 4】group\_start\_codeは、GOP層の開始同期コードを示すデータである。group\_of\_picture\_header\_present\_flagは、group\_of\_picture\_header内のデータ要素が有効であるか無効であることを示すデータである。time\_codeは、GOPの先頭ピクチャのシーケンスの先頭からの時間を示すタイムコードである。closed\_gopは、GOP内の画像が他のGOPから独立再生可能なことを示すフラグデータである。broken\_linkは、編集などのためにGOP内の先頭のBピクチャが正確に再生できないことを示すフラグデータである。

【0 3 2 5】extension\_and\_user\_data(1)関数は、extension\_and\_user\_data(0)関数と同じように、user\_data()関数によって定義されるデータ要素のみを履歴ストリームとして記述する。

【0 3 2 6】もし、履歴ストリーム中に、GOP層のスタ

ートコードを示すgroup\_start\_codeが存在しない場合には、これらのgroup\_of\_picture\_header()関数及びextension\_and\_user\_data(1)関数によって定義されるデータ要素は、履歴ストリーム中には記述されていない。その場合には、extension\_and\_user\_data(0)関数によって定義されたデータ要素の次に、picture\_header()関数によって定義されたデータ要素が履歴ストリームとして記述されている。

10 【0 3 2 7】picture\_header()関数によって定義されたデータ要素は、図 3 3 に示すように、picture\_start\_code、temporal\_reference、picture\_coding\_type、vbv\_delay、full\_pel\_forward\_vector、forward\_f\_code、full\_pel\_backward\_vector、backward\_f\_code、extra\_bit\_picture、及びextra\_information\_pictureである。

【0 3 2 8】具体的には、picture\_start\_codeは、ピクチャ層の開始同期コードを表すデータである。temporal\_referenceは、ピクチャの表示順を示す番号でGOPの先頭でリセットされるデータである。picture\_coding\_typeは、ピクチャタイプを示すデータである。vbv\_delayは、ランダムアクセス時の仮想バッファの初期状態を示すデータである。full\_pel\_forward\_vectorは、順方向動きベクトルの精度が整数単位か半画素単位かを示すデータである。forward\_f\_codeは、順方向動きベクトル探索範囲を表すデータである。full\_pel\_backward\_vectorは、逆方向動きベクトルの精度が整数単位か半画素単位かを示すデータである。backward\_f\_codeは、逆方向動きベクトル探索範囲を表すデータである。extra\_bit\_pictureは、後続する追加情報の存在を示すフラグである。このextra\_bit\_pictureが「1」の場合には、次にextra\_information\_pictureが存在し、extra\_bit\_pictureが「0」の場合には、これに続くデータが無いことを示している。extra\_information\_pictureは、規格において予約された情報である。

【0 3 2 9】picture\_header()関数によって定義されたデータ要素の次には、図 3 4 に示すようなpicture\_coding\_extension()関数によって定義されたデータ要素が、履歴ストリームとして記述されている。

40 【0 3 3 0】このpicture\_coding\_extension()関数によって定義されたデータ要素とは、extension\_start\_code、extension\_start\_code\_identifier、f\_code[0][0]、f\_code[0][1]、f\_code[1][0]、f\_code[1][1]、intra\_dc\_precision、picture\_structure、top\_field\_first、frame\_predictive\_frame\_dct、concealment\_motion\_vectors、q\_scale\_type、intra\_vlc\_format、alternate\_scan、repeat\_first\_field、chroma\_420\_type、progressive\_frame、composite\_display\_flag、v\_axis、field\_sequence、sub\_carrier、burst\_amplitude、及びsub\_carrier\_phaseから構成される。

50 【0 3 3 1】extension\_start\_codeは、ピクチャ層のエ

クステンションデータのスタートを示す開始コードである。extension\_start\_code\_identifierは、どの拡張データが送られるかを示すコードである。f\_code[0][0]は、フォワード方向の水平動きベクトル探索範囲を表すデータである。f\_code[0][1]は、フォワード方向の垂直動きベクトル探索範囲を表すデータである。f\_code[1][0]は、バックワード方向の水平動きベクトル探索範囲を表すデータである。f\_code[1][1]は、バックワード方向の垂直動きベクトル探索範囲を表すデータである。intra\_dc\_precisionは、DC係数の精度を表すデータである。

【0332】picture\_structureは、フレームストラクチャかフィールドストラクチャかを示すデータである。フィールドストラクチャの場合は、上位フィールドか下位フィールドかもあわせて示すデータである。top\_field\_firstは、フレームストラクチャの場合、最初のフィールドが上位か下位かを示すデータである。frame\_predictive\_frame\_dctは、フレーム・ストラクチャの場合、フレーム・モードDCTの予測がフレーム・モードだけであることを示すデータである。concealment\_motion\_vectorsは、イントラマクロブロックに伝送エラーを隠蔽するための動きベクトルがついていることを示すデータである。q\_scale\_typeは、線形量子化スケールを利用するか、非線形量子化スケールを利用するかを示すデータである。intra\_vlc\_formatは、イントラマクロブロックに、別の2次元VLCを使うかどうかを示すデータである。

【0333】alternate\_scanは、ジグザグスキャンを使うか、オルタネート・スキャンを使うかの選択を表すデータである。repeat\_first\_fieldは、2:3プルダウンの際に使われるデータである。chroma\_420\_typeは、信号フォーマットが4:2:0の場合、次のprogressive\_frameと同じ値、そうでない場合は0を表すデータである。progressive\_frameは、このピクチャが、順次走査できているかどうかを示すデータである。composite\_display\_flagは、ソース信号がコンポジット信号であったかどうかを示すデータである。v\_axisは、ソース信号が、PALの場合に使われるデータである。field\_sequenceは、ソース信号が、PALの場合に使われるデータである。sub\_carrierは、ソース信号が、PALの場合に使われるデータである。burst\_amplitudeは、ソース信号が、PALの場合に使われるデータである。sub\_carrier\_phaseは、ソース信号が、PALの場合に使われるデータである。

【0334】picture\_coding\_extension()関数によって定義されたデータエレメントの次には、re\_coding\_stream\_info()関数によって定義されたデータエレメントが履歴ストリームとして記述されている。このre\_coding\_stream\_info()関数は、主に履歴情報の組み合わせを記述する場合に用いられるものであり、その詳細について

は、図51を参照して後述する。

【0335】re\_coding\_stream\_info()関数によって定義されたデータエレメントの次には、extensions\_and\_user\_data(2)によって定義されたデータエレメントが、履歴ストリームとして記述されている。このextension\_and\_user\_data(2)関数は、図30に示したように、ビットストリーム中にエクステンションスタートコード(extension\_start\_code)が存在する場合には、extension\_data()関数によって定義されるデータエレメントが記述されている。このデータエレメントの次には、ビットストリーム中にユーザデータスタートコード(user\_data\_start\_code)が存在する場合には、user\_data()関数によって定義されるデータエレメントが記述されている。但し、ビットストリーム中にエクステンションスタートコード及びユーザデータスタートコードが存在しない場合にはextension\_data()関数及びuser\_data()関数によって定義されるデータエレメントはビットストリーム中には記述されていない。

【0336】extension\_data()関数は、図35に示すように、extension\_start\_codeを示すデータエレメントと、quant\_matrix\_extension()関数、copyright\_extension()関数、及びpicture\_display\_extension()関数によって定義されるデータエレメントとを、ビットストリーム中に履歴ストリームとして記述するための関数である。

【0337】quant\_matrix\_extension()関数によって定義されるデータエレメントは、図36に示すように、extension\_start\_code、extension\_start\_code\_identifier、quant\_matrix\_extension\_present\_flag、load\_intra\_quantiser\_matrix、intra\_quantiser\_matrix[64]、load\_non\_intra\_quantiser\_matrix、non\_intra\_quantiser\_matrix[64]、load\_chroma\_intra\_quantiser\_matrix、chroma\_intra\_quantiser\_matrix[64]、load\_chroma\_non\_intra\_quantiser\_matrix、及びchroma\_non\_intra\_quantiser\_matrix[64]である。

【0338】extension\_start\_codeは、この量子化マトリックスエクステンションのスタートを示す開始コードである。extension\_start\_code\_identifierは、どの拡張データが送られるかを示すコードである。quant\_matrix\_extension\_present\_flagは、この量子化マトリックスエクステンション内のデータエレメントが有効か無効かを示すためのデータである。load\_intra\_quantiser\_matrixは、イントラマクロブロック用の量子化マトリックスデータの存在を示すデータである。intra\_quantiser\_matrixは、イントラマクロブロック用の量子化マトリックスの値を示すデータである。

【0339】load\_non\_intra\_quantiser\_matrixは、非イントラマクロブロック用の量子化マトリックスデータの存在を示すデータである。non\_intra\_quantiser\_matrixは、非イントラマクロブロック用の量子化マトリックス

スの値を表すデータである。load\_chroma\_intra\_quantiser\_matrixは、色差イントラマクロブロック用の量子化マトリックス・データの存在を示すデータである。chroma\_intra\_quantiser\_matrixは、色差イントラマクロブロック用の量子化マトリックスの値を示すデータである。load\_chroma\_non\_intra\_quantiser\_matrixは、色差非イントラマクロブロック用の量子化マトリックス・データの存在を示すデータである。chroma\_non\_intra\_quantiser\_matrixは、色差非イントラマクロブロック用の量子化マトリックスの値を示すデータである。

【0340】copyright\_extension()関数によって定義されるデータエレメントは、図37に示すように、extension\_start\_code、extension\_start\_code\_identifier、copyright\_extension\_present\_flag、copyright\_flag、copyright\_identifier、original\_or\_copy、copyright\_number\_1、copyright\_number\_2、及びcopyright\_number\_3から構成される。

【0341】extension\_start\_codeは、コピーライトエクステンションのスタートを示す開始コードである。extension\_start\_code\_identifierは、このエクステンションデータが送られるかを示すコードである。copyright\_extension\_present\_flagは、このコピーライトエクステンション内のデータエレメントが有効か無効かを示すためのデータである。

【0342】copyright\_flagは、次のコピーライトエクステンション又はシーケンスエンドまで、符号化されたビデオデータに対してコピー権が与えられているか否かを示す。copyright\_identifierは、ISO/IEC JTC/SC29によって指定されたコピー権の登録機関を識別するためのデータである。original\_or\_copyは、ビットストリーム中のデータが、オリジナルデータであるかコピーデータであるかを示すデータである。copyright\_number\_1は、コピーライトナンバーのビット44から63を表わすデータである。copyright\_number\_2は、コピーライトナンバーのビット22から43を表わすデータである。copyright\_number\_3は、コピーライトナンバーのビット0から21を表わすデータである。

【0343】picture\_display\_extension()関数によって定義されるデータエレメントは、図38に示すように、extension\_start\_code\_identifier、frame\_center\_horizontal\_offset、frame\_center\_vertical\_offset等である。

【0344】extension\_start\_code\_identifierは、どの拡張データが送られるかを示すコードである。frame\_center\_horizontal\_offsetは、表示エリアの水平方向のオフセットを示すデータであって、number\_of\_frame\_center\_offsetsによって定義される数のオフセット値を定義することができる。frame\_center\_vertical\_offsetは、表示エリアを垂直方向のオフセットを示すデータであって、number\_of\_frame\_center\_offsetsによって定

義される数のオフセット値を定義することができる。

【0345】再び図27に戻って、extension\_and\_user\_data(2)関数によって定義されるデータエレメントの次には、picture\_data()関数によって定義されるデータエレメントが、履歴ストリームとして記述されている。但し、このpicture\_data()関数は、red\_bw\_flagが1ではないか、または、red\_bw\_indicatorが2以下である場合に存在する。このred\_bw\_flagとred\_bw\_indicatorは、re\_coding\_stream\_info()関数に記述されており、これらについては、図51と図52を参照して後述する。

【0346】picture\_data()関数によって定義されるデータエレメントは、図39に示すように、slice()関数によって定義されるデータエレメントである。このslice()関数によって定義されるデータエレメントはビットストリーム中に少なくとも1個記述されている。

【0347】slice()関数は、図40に示されるように、slice\_start\_code、slice\_quantiser\_scale\_code、intra\_slice\_flag、intra\_slice、reserved\_bits、extra\_bit\_slice、extra\_information\_slice、及びextra\_bit\_slice等のデータエレメントと、macroblock()関数によって定義されるデータエレメントを、履歴ストリームとして記述するための関数である。

【0348】slice\_start\_codeは、slice()関数によって定義されるデータエレメントのスタートを示すスタートコードである。slice\_quantiser\_scale\_codeは、このスライス層に存在するマクロブロックに対して設定された量子化ステップサイズを示すデータである。しかし、各マクロブロック毎に、quantiser\_scale\_codeが設定されている場合には、各マクロブロックに対して設定されたmacroblock\_quantiser\_scale\_codeのデータが優先して使用される。

【0349】intra\_slice\_flagは、ビットストリーム中にintra\_slice及びreserved\_bitsが存在するか否かを示すフラグである。intra\_sliceは、スライス層中にノンイントラマクロブロックが存在するか否かを示すデータである。スライス層におけるマクロブロックのいずれかがノンイントラマクロブロックである場合には、intra\_sliceは「0」となり、スライス層におけるマクロブロックの全てがノンイントラマクロブロックである場合には、intra\_sliceは「1」となる。reserved\_bitsは、7ビットのデータであって「0」の値を取る。extra\_bit\_sliceは、履歴ストリームとして追加の情報が存在することを示すフラグであって、次にextra\_information\_sliceが存在する場合には「1」に設定される。追加の情報が存在しない場合には「0」に設定される。

【0350】これらのデータエレメントの次には、macroblock()関数によって定義されたデータエレメントが、履歴ストリームとして記述されている。

【0351】macroblock()関数は、図41に示すように、macroblock\_escape、macroblock\_address\_increment



t、及びmacroblock\_quantiser\_scale\_code、及びmarker\_bit等のデータエレメントと、macroblock\_modes()関数、motion\_vectors(s)関数、及びcode\_block\_pattern()関数によって定義されたデータエレメントを記述するための関数である。

【0352】macroblock\_escapeは、参照マクロブロックと前のマクロブロックとの水平方向の差が34以上であるか否かを示す固定ビット列である。参照マクロブロックと前のマクロブロックとの水平方向の差が34以上の場合には、macroblock\_address\_incrementの値に33をプラスする。macroblock\_address\_incrementは、参照マクロブロックと前のマクロブロックとの水平方向の差を示すデータである。もし、このmacroblock\_address\_incrementの前にmacroblock\_escapeが1つ存在するのであれば、このmacroblock\_address\_incrementの値に33をプラスした値が、実際の参照マクロブロックと前のマクロブロックとの水平方向の差分を示すデータとなる。

【0353】macroblock\_quantiser\_scale\_codeは、各マクロブロック毎に設定された量子化ステップサイズであり、macroblock\_quantが"1"のときだけ存在する。各スライス層には、スライス層の量子化ステップサイズを示すslice\_quantiser\_scale\_codeが設定されているが、参照マクロブロックに対してmacroblock\_quantiser\_scale\_codeが設定されている場合には、この量子化ステップサイズを選択する。

【0354】macroblock\_address\_incrementの次には、macroblock\_modes()関数によって定義されるデータエレメントが記述されている。macroblock\_modes()関数は、図42に示すように、macroblock\_type、frame\_motion\_type、field\_motion\_type、dct\_type等のデータエレメントを、履歴ストリームとして記述するための関数である。

【0355】macroblock\_typeは、マクロブロックの符号化タイプを示すデータである。その詳細は、図45乃至図47を参照して後述する。

【0356】もし、macroblock\_motion\_forward又はmacroblock\_motion\_backwardが「1」であり、ピクチャ構造がフレームであり、さらにframe\_pred\_frame\_dctが「0」である場合には、macroblock\_typeを表わすデータエレメントの次にframe\_motion\_typeを表わすデータエレメントが記述されている。尚、このframe\_pred\_frame\_dctは、frame\_motion\_typeがビットストリーム中に存在するか否かを示すフラグである。

【0357】frame\_motion\_typeは、フレームのマクロブロックの予測タイプを示す2ビットのコードである。予測ベクトルが2個でフィールドベースの予測タイプであれば「00」であって、予測ベクトルが1個でフィールドベースの予測タイプであれば「01」であって、予測ベクトルが1個でフレームベースの予測タイプであれば「10」であって、予測ベクトルが1個でディアルプ

ライムの予測タイプであれば「11」である。

【0358】frame\_motion\_typeを記述する条件が満足されない場合には、macroblock\_typeを表わすデータエレメントの次にfield\_motion\_typeを表わすデータエレメントが記述されている。

【0359】field\_motion\_typeは、フィールドのマクロブロックの動き予測を示す2ビットのコードである。予測ベクトルが1個でフィールドベースの予測タイプであれば「01」であって、予測ベクトルが2個で18×8マクロブロックベースの予測タイプであれば「10」であって、予測ベクトルが1個でディアルプライムの予測タイプであれば「11」である。

【0360】もし、ピクチャ構造がフレームで、frame\_pred\_frame\_dctがframe\_motion\_typeがビットストリーム中に存在することを示し、且つ、frame\_pred\_frame\_dctがdct\_typeがビットストリーム中に存在することを示している場合には、macroblock\_typeを表わすデータエレメントの次にdct\_typeを表わすデータエレメントが記述されている。尚、dct\_typeは、DCTがフレームDCTモードか、フィールドDCTモードかを示すデータである。

【0361】再び図41に戻って、もし、参照マクロブロックが前方予測マクロブロックであるか、又は参照マクロブロックがイントラマクロブロックであって且つコンシール処理のマクロブロックのいずれかの場合には、motion\_vectors()関数によって定義されるデータエレメントが記述される。また、参照マクロブロックが後方予測マクロブロックである場合には、motion\_vectors(1)関数によって定義されるデータエレメントが記述される。尚、motion\_vectors(0)関数は、第1番目の動きベクトルに関するデータエレメントを記述するための関数であって、motion\_vectors(1)関数は、第2番目の動きベクトルに関するデータエレメントを記述するための関数である。

【0362】motion\_vectors(s)関数は、図43に示されるように、動きベクトルに関するデータエレメントを記述するための関数である。

【0363】もし、動きベクトルが1個でディアルプライム予測モードを使用していない場合には、motion\_vertical\_field\_select[0][s]とmotion\_vector(0,s)によって定義されるデータエレメントが記述される。

【0364】このmotion\_vertical\_field\_select[r][s]は、第1番目の動きベクトル（前方又は後方のどちらのベクトルであっても良い）が、ボトムフィールドを参照して作られたベクトルであるかトップフィールドを参照して作られたベクトルであるかを示すフラグである。この指標“r”は、第1番目のベクトル又は第2番目のベクトルのいずれのベクトルであるかを示す指標であって、“s”は、予測方向が前方又は後方予測のいずれであるかを示す指標である。

【0365】motion\_vector(r,s)関数は、図44に示さ

れるように、motion\_code[r][s][t]に関するデータ列と、motion\_residual[r][s][t]に関するデータ列と、dmvector[t]を表わすデータとを記述するための関数である。

【0 3 6 6】motion\_code[r][s][t]は、動きベクトルの大きさを-16～+16の範囲で表わす可変長のデータである。motion\_residual[r][s][t]は、動きベクトルの残差を表わす可変長のデータである。よって、このmotion\_code[r][s][t]とmotion\_residual[r][s][t]との値によって詳細な動きベクトルを記述することができる。dmvector[t]は、ディユアルプライム予測モードのときに、一方のフィールド（例えばボトムフィールドに対してトップフィールドを一方のフィールドとする）における動きベクトルを生成するために、時間距離に応じて既存の動きベクトルがスケールされると共に、トップフィールドとボトムフィールドとのライン間の垂直方向のずれを反映させるために垂直方向に対して補正を行うデータである。この指標“r”は、第1番めのベクトル又は第2番めのベクトルのいずれのベクトルであるかを示す指標であって、“s”は、予測方向が前方又は後方予測のいずれであるかを示す指標である。“s”は、動きベクトルが垂直方向の成分であるか水平方向の成分であるかを示すデータである。

【0 3 6 7】図 4 4 に示されmotion\_vector(r, s)関数によって、まず、水平方向のmotion\_coder[r][s][0]を表わすデータ列が、履歴ストリームとして記述される。motion\_residual[0][s][t]及びmotion\_residual[1][s][t]の双方のビット数は、f\_code[s][t]で示されるので、f\_code[s][t]が1でない場合には、motion\_residual[r][s][t]がビットストリーム中に存在することを示すことになる。水平方向成分のmotion\_residual[r][s][0]が「1」でなくて、水平方向成分のmotion\_code[r][s][0]が「0」でないということは、ビットストリーム中にmotion\_residual[r][s][0]を表わすデータエレメントが存在し、動きベクトルの水平方向成分が存在するということを示している。その場合には、水平方向成分のmotion\_residual[r][s][0]を表わすデータエレメントが記述されている。

【0 3 6 8】続いて、垂直方向のmotion\_coder[r][s][1]を表わすデータ列が、履歴ストリームとして記述される。同じようにmotion\_residual[0][s][t]及びmotion\_residual[1][s][t]の双方のビット数は、f\_code[s][t]で示されるので、f\_code[s][t]が1でない場合には、motion\_residual[r][s][t]がビットストリーム中に存在することを表わすことになる。motion\_residual[r][s][1]が「1」でなくて、motion\_code[r][s][1]が「0」でないということは、ビットストリーム中にmotion\_residual[r][s][1]を表わすデータエレメントが存在し、動きベクトルの垂直方向成分が存在するということを示している。その場合には、垂直方向成分のmo

tion\_residual[r][s][1]を表わすデータエレメントが記述されている。

【0 3 6 9】次に、図 4 5 乃至図 4 7 を参照して、macroblock\_typeについて説明する。macroblock\_typeは、macroblock\_quant、dct\_type\_flag、macroblock\_motion\_forward、及びmacroblock\_motion\_backwardなどのフラグから生成された可変長データである。macroblock\_quantは、マクロブロックに対して量子化ステップサイズを設定するためのmacroblock\_quantiser\_scale\_codeが設定されているか否かを示すフラグであって、ビットストリーム中にmacroblock\_quantiser\_scale\_codeが存在する場合には、macroblock\_quantは「1」の値を取る。

【0 3 7 0】dct\_type\_flagは、参照マクロブロックがフレームDCT又はフィールドDCTで符号化されているかを示すdct\_typeが存在するか否かを示すためのフラグ（言い換えるとDCTされているか否かを示すフラグ）であって、ビットストリーム中にdct\_typeが存在する場合には、このdct\_type\_flagは「1」の値を取る。macroblock\_motion\_forwardは、参照マクロブロックが前方予測されているか否かを示すフラグであって、前方予測されている場合には「1」の値を取る。macroblock\_motion\_backwardは、参照マクロブロックが後方予測されているか否かを示すフラグであって、後方予測されている場合には「1」の値を取る。

【0 3 7 1】なお、可変長フォーマットにおいては、伝送するビットレートを減少させるために、履歴情報を削減することができる。

【0 3 7 2】すなわち、macroblock\_typeとmotion\_vectors()は転送するが、quantiser\_scale\_codeを転送しない場合には、slice\_quantiser\_scale\_codeを”0 0 0 0”とすることで、ビットレートを減少させることができる。

【0 3 7 3】また、macroblock\_typeのみ転送し、motion\_vectors()、quantiser\_scale\_code、およびdct\_typeを転送しない場合には、macroblock\_typeとして、“not coded”を使用することで、ビットレートを減少することができる。

【0 3 7 4】さらにまた、picture\_coding\_typeのみ転送し、slice()以下の情報は全て転送しない場合には、slice\_start\_codeを持たないpicture\_data()を使用することで、ビットレートを減少させることができる。

【0 3 7 5】以上においては、user\_data内の2 3 ビットの連続する”0”が出ないようにする場合に、2 2 ビット毎に”1”を挿入するようにしたが、2 2 ビット毎でなくてもよい。また、連続する”0”の個数を数えて”1”を挿入するのではなく、Byte\_alignを調べて挿入するようにすることも可能である。

【0 3 7 6】さらに、MPEGにおいては、2 3 ビットの連続する”0”の発生を禁止しているが、実際には、バイトの先頭から2 3 ビット連続する場合だけが問題とさ

れ、バイトの先頭ではなく、途中から 0 が 2 3 ビット連続する場合は、問題とされない。従って、例えば 2 4 ビット毎に、LSB 以外の位置に " 1 " を挿入するようにしてもよい。

【0 3 7 7】また、以上においては、履歴情報を、video elementary stream に近い形式にしたが、packetized elementary stream や transport stream に近い形式にしてもよい。また、Elementary Stream の user\_data の場所を、picture\_data の前としたが、他の場所にすることもできる。

【0 3 7 8】図 4 のトランスコーディングシステム 1 においては、4 世代分の符号化パラメータを履歴情報として後段に出力するようにしたが、実際には、履歴情報の全てが必要となるわけではなく、アプリケーション毎に必要な履歴情報は異なってくる。また、実際の伝送路あるいは記録媒体（伝送メディア）には、容量に制限があり、圧縮しているとはいえ、全ての履歴情報を伝送するようにすると、容量的に負担となり、結果的に画像ビットストリームのビットレートを抑圧してしまい、履歴情報伝送の有効性が損なわれることになる。

【0 3 7 9】そこで、履歴情報として伝送する項目の組み合わせを記述する記述子を履歴情報に組み込んで後段に送信するようにし、全ての履歴情報を伝送するのではなく、様々なアプリケーションに対応した情報を伝送するようにすることができる。図 4 8 は、このような場合のトランスコーディングシステム 1 の構成例を表している。

【0 3 8 0】図 4 8 において、図 4 における場合と対応する部分には同一の符号を付してあり、その説明は適宜省略する。図 4 8 の構成例においては、ヒストリ情報分離装置 1 1 5 と符号化装置 1 1 6 の間、及びヒストリエンコーディング装置 1 1 7 と符号化装置 1 1 6 の間に、符号化パラメータ選択回路 5 0 1 が挿入されている。

【0 3 8 1】符号化パラメータ選択回路 5 0 1 は、ヒストリ情報分離装置 1 1 5 が出力するベースバンドビデオ信号から符号化パラメータを算出する符号化パラメータ算出部 5 1 2、ヒストリ情報分離装置 1 1 5 が出力する、このトランスコーディングシステム 1 において、符号化するのに最適と判定された符号化パラメータ（例えば、第 2 世代の符号化パラメータ）に関する情報から、符号化パラメータと記述子（red\_bw\_flag, red\_bw\_indicator）（図 5 2 を参照して後述する）を分離する組合せ記述子分離部 5 1 1、並びに符号化パラメータ算出部 5 1 2 が出力する符号化パラメータと、組合せ記述子分離部 5 1 1 が出力する符号化パラメータのうち、いずれか一方を、組合せ記述子分離部 5 1 1 で分離された記述子に対応して選択し、符号化装置 1 1 6 に出力するスイッチ 5 1 3 を有している。その他の構成は、図 4 における場合と同様である。

【0 3 8 2】ここで、履歴情報として伝送する項目の組

み合わせについて説明する。履歴情報は、分類すると、picture 単位の情報と、macroblock 単位の情報に分けることができる。slice 単位の情報は、それに含まれる macroblock の情報を収集することで得ることができ、GOP 単位の情報は、それに含まれる picture 単位の情報を収集することで得ることができる。

【0 3 8 3】picture 単位の情報は、1 フレーム毎に 1 回伝送されるだけなので、情報伝送に占めるビットレートは、それほど大きくはない。これに対して、macroblock 単位の情報は、各 macroblock 毎に伝送されるため、例えば 1 フレームの走査線数が 5 2 5 本で、フィールドレートが 6 0 フィールド／秒のビデオシステムの場合、1 フレームの画素数を  $720 \times 480$  とすると、macroblock 単位の情報は、1 フレームあたり  $1350 (= (720 / 16) \times (480 / 16))$  回伝送することが必要となる。このため、履歴情報の相当の部分が macroblock 毎の情報で占められることになる。そこで、履歴情報としては、少なくとも picture 単位の情報は常に伝送するが、macroblock 単位の情報は、アプリケーションに応じて選択して伝送するようにすることで、伝送する情報量を抑制することができる。

【0 3 8 4】履歴情報として転送される macroblock 単位の情報には、例えば num\_coef\_bits, num\_mv\_bits, num\_other\_bits, q\_scale\_code, q\_scale\_type, motion\_type, mv\_vert\_field\_sel[], mv[], mb\_mfwd, mb\_mbwd, mb\_pattern, coded\_block\_pattern, mb\_intra, slice\_start, dct\_type, mb\_quant, skipped\_mb などがある。これらは、macroblock rate information の要素を用いて表現されたものである。

【0 3 8 5】num\_coef\_bits は、macroblock の符号量のうち、DCT 係数に要した符号量を表す。num\_mv\_bits は、macroblock の符号量のうち、動きベクトルに要した符号量を表す。num\_other\_bits は、macroblock の符号量のうち、num\_coef\_bits 及び num\_mv\_bits 以外の符号量を表す。

【0 3 8 6】q\_scale\_code は、macroblock に適用された q\_scale\_code を表す。motion\_type は、macroblock に適用された動きベクトルの type を表す。mv\_vert\_field\_sel[] は、macroblock に適用された動きベクトルの field select を表す。

【0 3 8 7】mv[] は、macroblock に適用された動きベクトルを表す。mb\_mfwd は、macroblock の予測モードが前方予測であることを示すフラグである。mb\_mbwd は、macroblock の予測モードが後方向予測であることを示すフラグである。mb\_pattern は、macroblock の DCT 係数の非 0 のものの有無を示すフラグである。

【0 3 8 8】coded\_block\_pattern は、macroblock の DCT 係数の非 0 のものの有無を DCT ブロック毎に示すフラグである。mb\_intra は、macroblock が intra\_macro かそうでないかを示すフラグである。slice\_start は、macrobl

ockがsliceの先頭であるか否かを示すフラグである。dct\_typeは、macroblockがfield\_dctかflame\_dctかを示すフラグである。

【0389】mb\_quantは、macroblockがquantiser\_scale\_codeを伝送するか否かを示すフラグである。skipped\_mbは、macroblockがskipped macroblockであるか否かを示すフラグである。

【0390】これらの項目は、常に全て必要であるわけではなく、アプリケーションに応じて必要となる項目が変化する。例えば、num\_coef\_bitsやslice\_startといった項目は、再エンコードした際のビットストリームをできる限り元の形に戻したいというtransparentという要求を有するアプリケーションにおいて必要となる。換言すれば、ビットレートを変更するようなアプリケーションにおいては、これらの項目は必要ではない。また、非常に伝送路の制限が厳しい場合には、各ピクチャの符号化タイプが判るだけでもよいようなアプリケーションも存在する。このような状況から、履歴情報を伝送する項目の組み合わせの例として、例えば図49に示すような組み合わせが考えられる。

【0391】図49において、各組み合わせの中の項目に対応する値「2」は、その情報が存在し、利用可能であることを意味し、「0」は、その情報が存在しないことを意味する。「1」は、他の情報の存在を補助する目的のため、あるいは、構文上存在するが、元のビットストリーム情報とは関係がないなど、その情報自身には意味がないことを表している。例えば、slice\_startは、履歴情報を伝送する際のsliceの先頭のmacroblockにおいて、「1」になるが、本来のビットストリームに対して、sliceが必ずしも同一位置関係にあるわけではない場合には、履歴情報としては無意味になる。

【0392】図49の例においては、(num\_coef\_bits, num\_mv\_bits, num\_other\_bits), (q\_scale\_code, q\_scale\_type), (motion\_type, mv\_vert\_field\_sel[], mv[] []), (mb\_mfwd, mb\_mbwd), (mb\_pattern), (coded\_block\_pattern), (mb\_intra), (slice\_start), (dct\_type), (mb\_quant), (skipped\_mb)の各項目の有無により、組み合わせ1乃至組み合わせ5の5つの組み合わせが用意されている。

【0393】組み合わせ1は、完全にtransparentなビットストリームを再構成することを目的とした組み合わせである。この組み合わせによれば、発生符号量情報を用いることによる精度の高いトランスコーディングが実現できる。組み合わせ2も、完全にtransparentなビットストリームを再構成することを目的とした組み合わせである。組み合わせ3は、完全にtransparentなビットストリームを再構成することはできないが、視覚的にはtransparentなビットストリームを再構成できるようにするための組み合わせである。組み合わせ4は、transparentという観点からは組み合わせ3よりも劣るが、

視覚上問題がないビットストリームの再構成ができる組み合わせである。組み合わせ5は、transparentという観点からは組み合わせ4よりも劣るが、少ない履歴情報でビットストリームの完全ではない再構成ができる組み合わせである。

【0394】これらの組み合わせのうち、組み合わせの番号の数字が小さいものほど、機能的には上位であるが、履歴を転送するのに必要となる容量が多くなる。従って、想定するアプリケーションと履歴に使用できる容量を考慮することによって、伝送する組み合わせを決定する必要がある。

【0395】次に、図50のフローチャートを参照して、図48のトランスコーディングシステム1の動作について説明する。ステップS41において、トランスコーディングシステム1の復号装置102は、入力されたビットストリームを復号し、そのビットストリームを符号化する際に使用された符号化パラメータ(4th)を抽出し、その符号化パラメータ(4th)をヒストリ情報多重化装置103に出力するとともに、復号したビデオデータをやはりヒストリ情報多重化装置103に出力する。ステップS42において、復号装置102はまた、入力されたビットストリームからuser\_dataを抽出し、ヒストリデコーディング装置104に出力する。ヒストリデコーディング装置104は、ステップS43において、入力されたuser\_dataから、組み合わせ情報(記述子)を抽出し、さらにそれを用いて、履歴情報としての符号化パラメータ(1st, 2nd, 3rd)を抽出し、ヒストリ情報多重化装置103に出力する。

【0396】ヒストリ情報多重化装置103は、ステップS44において、ステップS41で取り出された復号装置102から供給される現在の符号化パラメータ(4th)と、ステップS43でヒストリデコーディング装置104が出力した過去の符号化パラメータ(1st, 2nd, 3rd)とを、復号装置102から供給されるベースバンドのビデオデータに、図7または図35に示すようなフォーマットに従って多重化し、ヒストリ情報分離装置115に出力する。

【0397】ヒストリ情報分離装置115は、ステップS45において、ヒストリ情報多重化装置103より供給されたベースバンドのビデオデータから符号化パラメータを抽出し、その中から今回の符号化に最も適している符号化パラメータ(例えば、第2世代の符号化パラメータ)を選択し、記述子とともに、組合せ記述子分離部511に出力する。また、ヒストリ情報分離装置115は、今回の符号化に最適と判定された符号化パラメータ以外の符号化パラメータ(例えば、最適な符号化パラメータが第2世代の符号化パラメータであると判定された場合には、それ以外の第1世代、第3世代、及び第4世代の符号化パラメータ)をヒストリエンコーディング装置117に出力する。ヒストリエンコーディング装置1

17は、ヒストリ情報分離装置115より入力された符号化パラメータをステップS46において、user\_dataに記述し、そのuser\_data(converted\_history\_stream())を符号化装置116に出力する。

【0398】符号化パラメータ選択回路501の組合せ記述子分離部511は、ヒストリ情報分離装置115より供給されたデータから、符号化パラメータと記述子を分離し、符号化パラメータ(2nd)をスイッチ513の一方の接点に供給する。スイッチ513の他方の接点には、符号化パラメータ算出部512が、ヒストリ情報分離装置115が出力するベースバンドのビデオデータから、符号化パラメータを算出し、供給している。スイッチ513は、ステップS48において、組合せ記述子分離部511が出力した記述子に対応して、組合せ記述子分離部511が出力した符号化パラメータ、または符号化パラメータ算出部512が出力した符号化パラメータのいずれかを選択し、符号化装置116に出力する。すなわち、スイッチ513では、組合せ記述子分離部511から供給された符号化パラメータが有効である場合には、組合せ記述子分離部511が出力する符号化パラメータが選択されるが、組合せ記述子分離部511が出力する符号化パラメータが無効であると判定された場合には、符号化パラメータ算出部512がベースバンドビデオを処理することで算出した符号化パラメータが選択される。この選択は、伝送メディアの容量に対応して行われる。

【0399】符号化装置116は、ステップS49において、スイッチ513から供給された符号化パラメータに基づいて、ヒストリ情報分離装置115より供給されたベースバンドビデオ信号を符号化する。また、ステップS50において、符号化装置116は、符号化したビットストリームに、ヒストリエンコーディング装置117より供給されたuser\_dataを多重化し、出力する。

【0400】このようにして、各履歴によって得られる符号化パラメータの組み合わせが異なっているような場合でも、支障なくトランスコーディングすることが可能となる。

【0401】このように、履歴情報は、図18に示したように、ビデオストリームのuser\_data()関数の一種としてのhistory\_stream() (より正確には、converted\_history\_stream())で伝送される。そのhistory\_stream()のシンタックスは、図27に示した通りである。履歴情報の項目の組み合わせを表す記述子(red\_bw\_flag, red\_bw\_indicator)、およびMPEGのストリームではサポートされていない項目(num\_other\_bits, num\_mv\_bits, num\_coef\_bits)は、この図27の中のre\_coding\_stream\_info()関数により伝送される。

【0402】re\_coding\_stream\_info()関数は、図51に示すように、user\_data\_start\_code, re\_coding\_stream\_info\_ID, red\_bw\_flag, red\_bw\_indicator, marker\_

bit, num\_other\_bits, num\_mv\_bits, num\_coef\_bitsなどのデータエレメントより構成される。

【0403】user\_data\_start\_codeは、user\_dataが開始することを表すスタートコードである。re\_coding\_stream\_info\_IDは、16ビットの整数であり、re\_coding\_stream\_info()関数の識別のために用いられる。その値は、具体的には、"1001 00011110 1100" (0x91ec)とされる。

【0404】red\_bw\_flagは、1ビットのフラグであり、履歴情報が全ての項目を伝送する場合には0とされ、このフラグの値が1である場合、このフラグに続くred\_bw\_indicatorを調べることにより、図49に示した5個の組み合わせのうち、どの組み合わせで項目が送られているのかを決定することができる。

【0405】red\_bw\_indicatorは、2ビットの整数であり、項目の組み合わせを図52に示すように記述する。

【0406】即ち、図49に示した5つの組み合わせのうち、組み合わせ1の場合、red\_bw\_flagは0とされ、組み合わせ2乃至組み合わせ5のとき、red\_bw\_flagは1とされる。これに対して、red\_bw\_indicatorは、組み合わせ2の場合0とされ、組み合わせ3の場合1とされ、組み合わせ4の場合2とされ、組み合わせ5の場合3とされる。

【0407】従って、red\_bw\_indicatorは、red\_bw\_flagが1の場合に(組み合わせ2乃至組み合わせ5の場合に)規定される。

【0408】さらに、図51に示すように、red\_bw\_flagが0である場合(組み合わせ1の場合)、マクロブロック毎に、marker\_bit, num\_other\_bits, num\_mv\_bits, num\_coef\_bitsが記述される。これら4つのデータエレメントは、組み合わせ2乃至組み合わせ5の場合(red\_bw\_flagが1の場合)規定されない。

【0409】図39に示したように、picture\_data()関数は、1個以上のslice()関数から構成される。しかしながら、組み合わせ5の場合、picture\_data()関数を含めて、それ以下のシンタックス要素は伝送されない(図49)。この場合、履歴情報は、picture\_typeなどのpicture単位の情報の伝送を意図したものとなる。

【0410】組み合わせ1乃至組み合わせ4の場合、図40に示したslice()関数が存在する。しかしながら、このslice()関数によって決定されるsliceの位置情報と、元のビットストリームのsliceの位置情報は、履歴情報の項目の組み合わせに依存する。組み合わせ1または組み合わせ2の場合、履歴情報の元となったビットストリームのsliceの位置情報と、slice()関数によって決定されるsliceの位置情報とは、同一である必要がある。

【0411】図41に示すmacroblock()関数のシンタックス要素は、履歴情報の項目の組み合わせに依存する。macroblock\_escape, macroblock\_address\_increment, m

acrobblock\_modes()関数は、常に存在する。しかしながら、macroblock\_escapeとmacroblock\_address\_incrementの情報としての有効性は、組み合わせによって決定される。履歴情報の項目の組み合わせが、組み合わせ1または組み合わせ2の場合、元のビットストリームのskip ped\_mb情報と同じものが伝送される必要がある。

【0 4 1 2】組み合わせ4の場合、motion\_vectors()関数は存在しない。組み合わせ1乃至組み合わせ3の場合、macroblock\_modes()関数のmacroblock\_typeによって、motion\_vectors()関数の存在が決定される。組み合わせ3または組み合わせ4の場合には、coded\_block\_pattern()関数は存在しない。組み合わせ1と組み合わせ2の場合、macroblock\_modes()関数のmacroblock\_typeによって、coded\_block\_pattern()関数の存在が決定される。

【0 4 1 3】図 4 2 に示したmacroblock\_modes()関数のシンタックス要素は、履歴情報の項目の組み合わせに依存する。macroblock\_typeは、常に存在する。組み合わせが組み合わせ4である場合、flame\_motion\_type, field\_motion\_type, dct\_typeは存在しない。

【0 4 1 4】macroblock\_typeより得られるパラメータの情報としての有効性は、履歴情報の項目の組み合わせによって決定される。

【0 4 1 5】履歴情報の項目の組み合わせが組み合わせ1または組み合わせ2である場合、macroblock\_quantは、元のビットストリームと同じである必要がある。組み合わせ3または組み合わせ4の場合、macroblock\_quantは、macroblock()関数内のquantiser\_scale\_codeの存在を表し、元のビットストリームと同じである必要はない。

【0 4 1 6】組み合わせが組み合わせ1乃至組み合わせ3である場合、macroblock\_motion\_forwardとmacroblock\_motion\_backwardは、元のビットストリームと同一である必要がある。組み合わせが組み合わせ4または組み合わせ5である場合、その必要はない。

【0 4 1 7】組み合わせが組み合わせ1または組み合わせ2である場合、macroblock\_patternは、元のビットストリームと同一である必要がある。組み合わせ3の場合、macroblock\_patternは、dct\_typeの存在を示すのに用いられる。組み合わせが組み合わせ4である場合、組み合わせ1乃至組み合わせ3における場合のような関係は成立しない。

【0 4 1 8】履歴情報の項目の組み合わせが組み合わせ1乃至組み合わせ3の場合、macroblock\_intraは、元のビットストリームと同一である必要がある。組み合わせ4の場合には、その限りでない。

【0 4 1 9】図 2 7 のhistory\_stream()は、履歴情報を可変長とする場合のシンタックスであるが、図 2 0 乃至図 2 6 に示すように、固定長のシンタックスとする場合、固定長の履歴情報内に、伝送される項目中のどれが

有効であることを示す情報としての記述子(red\_bw\_flagとred\_bw\_indicator)をベースバンド画像に重畳し、伝送するようにする。その結果、この記述子を調べることにより、フィールドとして存在するが、その内容は無効であるといった判断をすることが可能となる。

【0 4 2 0】このため、図 2 4 に示すように、re\_coding\_stream\_informationとして、user\_data\_start\_code, re\_coding\_stream\_info\_ID, red\_bw\_flag, red\_bw\_indicator, marker\_bitが配置されている。それぞれの意味は、図 5 1 における場合と同様である。

【0 4 2 1】このように履歴として伝送する符号化パラメータの要素をアプリケーションに応じた組み合わせで伝送するようにすることで、アプリケーションに応じた履歴を適当なデータ量で伝送するようにすることができる。

【0 4 2 2】以上のように、履歴情報を可変長符号として伝送する場合、re\_coding\_stream\_info()関数は、図 5 1 に示すように構成され、図 2 7 に示すように、history\_stream()関数の一部として伝送される。これに対して、履歴情報を固定長符号として伝送する場合には、図 2 4 に示したように、history\_stream()関数の一部として、re\_coding\_stream\_information()が伝送される。図 2 4 の例では、re\_coding\_stream\_informationとして、user\_data\_start\_code, re\_coding\_stream\_info\_ID, red\_bw\_flag, red\_bw\_indicatorが伝送される。

【0 4 2 3】また、図 4 8 のヒストリ情報多重化装置 1 0 3 が出力するベースバンドの信号中における履歴情報の伝送のために、図 5 3 に示すようなRe\_Coding information Bus macroblock formatが規定される。このマクロブロックは、1 6 × 1 6 (= 2 5 6) ビットで構成される。そして、そのうちの図 5 3 において上から3行目と4行目に示す3 2 ビットが、picrate\_elementとされる。このpicrate\_elementには、図 5 4 乃至図 5 6 に示すPicture rate elementsが記述される。図 5 4 の上から2行目に1 ビットのred\_bw\_flagが規定されており、また、3行目に3 ビットのred\_bw\_indicatorが規定されている。即ち、これらのフラグred\_bw\_flag, red\_bw\_indicatorは、図 5 3 のpicrate\_elementとして伝送される。

【0 4 2 4】図 5 3 のその他のデータについて説明すると、SRIB\_sync\_codeは、このフォーマットのマクロブロックの最初の行が左詰めにアライメントされていることを表すコードであり、具体的には、“1 1 1 1 1”に設定される。fr\_fl\_SRIBは、picture\_structureがフレームピクチャ構造の場合(その値が“1 1”である場合)、1に設定され、Re\_Coding Information Bus macroblockが1 6 ラインを超えて伝送されることを表し、picture\_structureがフレーム構造ではない場合、0に設定され、Re\_Coding Information Busが1 6 ラインを超えて伝送されることを意味する。この機構により、Re\_C

oding Information Busが、空間的かつ時間的にデコードされたビデオフレームまたはフィールドの対応する画素にロックされる。

【0 4 2 5】SRIB\_top\_field\_firstは、元のビットストリームに保持されているtop\_field\_firstと同じ値に設定され、関連するビデオのRe\_Coding Information Busの時間的アライメントをrepeat\_first\_fieldとともに表している。SRIB\_repeat\_first\_fieldは、元のビットストリームに保持されているrepeat\_first\_fieldと同じ値に設定される。first fieldのRe\_Coding Information Busの内容は、このフラグに示されるように繰り返される必要がある。

【0 4 2 6】422\_420\_chromaは、元のビットストリームが4 : 2 : 2または4 : 2 : 0のいずれであるかを表す。その値の0は、ビットストリームが4 : 2 : 0であり、色差信号のアップサンプリングが、4 : 2 : 2のビデオが出力されるように行われたことを表す。その値の0は、色差信号のフィルタリング処理が実行されていないことを表す。

【0 4 2 7】rolling\_SRIB\_mb\_refは、1 6 ビットのモジュロ65521を表し、この値は、毎マクロブロック毎にインクリメントされる。この値は、フレームピクチャ構造のフレームに渡って連続している必要がある。さもなくば、この値は、フィールドに渡って連続している必要がある。この値は、0 から65520の間の所定の値に初期化される。これにより、レコーダのシステムに、ユニークなRe\_Coding Information Busの識別子を組み込むことが許容される。

【0 4 2 8】Re\_Coding Information Bus macroblockのその他のデータの意味は、上述した通りであるので、ここでは省略する。

【0 4 2 9】図 5 7 に示すように、図 5 3 の 2 5 6 ビットのRe\_Coding Information Busのデータは、1 ビットずつ、色差データのLSBであるCb[0][0], Cr[0][0], Cb[1][0], Cr[1][0]に配置される。図 5 7 に示すフォーマットにより、4 ビットのデータを送ることができるので、図 5 3 の 2 5 6 ビットのデータは、図 5 7 のフォーマットを6 4 (= 2 5 6 / 4) 個送ることで伝送することができる。

【0 4 3 0】本発明のトランスコーディングシステムによれば、過去の符号化処理において生成された符号化パラメータを、現在の符号化処理において再利用するようにしているので、復号処理及び符号化処理を繰り返したとしても画質劣化が発生しない。つまり、復号処理及び符号化処理の繰り返しによる画質劣化の蓄積を低減することができる。

【0 4 3 1】本発明のトランスコーディングシステムによれば、過去の符号化処理において生成された符号化パラメータを、現在の符号化処理において生成された符号化ストリームのユーザデータエリアに記述するように

し、生成されたビットストリームは、MPEG規格に準じた符号化ストリームであるので、既存のどのデコーダでも復号処理を行うことができる。さらには、本発明のトランスコーディングシステムによれば、過去の符号化処理における符号化パラメータを伝送するために専用線のようなものを設ける必要がないので、従来のデータストリーム伝送環境をそのまま使用して、過去の符号化パラメータを伝送することができる。

【0 4 3 2】本発明のトランスコーディングシステムによれば、過去の符号化処理において生成された符号化パラメータを、選択的に現在の符号化処理において生成された符号化ストリーム中に記述するようにしているので、出力されるビットストリームのビットレートを極端に上げることなく、過去の符号化パラメータを伝送することができる。

【0 4 3 3】本発明のトランスコーディングシステムによれば、過去の符号化パラメータと現在の符号化パラメータの中から、現在の符号化処理に最適な符号化パラメータを選択して符号化処理を行うようにしているので、復号処理及び符号化処理を繰り返したとしても、画質劣化が蓄積されることはない。

【0 4 3 4】本発明のトランスコーディングシステムによれば、過去の符号化パラメータの中から、ピクチャタイプに応じて現在の符号化処理に最適な符号化パラメータを選択して符号化処理を行うようにしているので、復号処理及び符号化処理を繰り返したとしても、画質劣化が蓄積されることはない。

【0 4 3 5】本発明のトランスコーディングシステムによれば、過去の符号化パラメータに含まれるピクチャタイプに基づいて、過去の符号化パラメータを再利用するか否かを決定しているので、最適な符号化処理を行うことができる。

【0 4 3 6】なお、カウンタ1 0 1、カウンタ1 1 3、カウンタ3 6 2、およびカウンタ3 6 4は、バイナリカウンタであるとして説明したが、グレイコード（巡回2進符号）のカウンタでもよい。

【0 4 3 7】また、ビデオ復号システム1 1または復号装置1 0 2は、ベースバンドデジタルビデオ信号を出力し、ビデオ符号化システム1 2または符号化装置1 1 6は、ベースバンドデジタルビデオ信号を入力するとして説明したが、ビデオ復号システム1 1または復号装置1 0 2は、アナログビデオ信号を出力し、ビデオ符号化システム1 2または符号化装置1 1 6は、アナログビデオ信号を入力するようにしてもよい。

【0 4 3 8】なお、カウンタ値は、画像に多重化するとして説明したが、画像に関連付けられている信号、例えば、オーディオ信号に多重化するようにしてもよい。

【0 4 3 9】また、上記各処理を行うコンピュータプログラムは、磁気ディスク、光ディスク、光磁気ディスク、半導体メモリなどの記録媒体に記録して提供するほ

か、インターネット、デジタル衛星などのネットワークを介して伝送し、ユーザの記録媒体に記録させることで提供することができる。

#### 【0 4 4 0】

【発明の効果】請求項 1 に記載のストリーム生成装置、請求項 1 3 に記載のストリーム生成方法および請求項 1 4 に記載の記録媒体によれば、第 1 のストリームの過去の符号化処理における符号化履歴が検出され、第 1 のストリームの画像の不連続が検出され、符号化履歴の検出結果および画像の不連続の検出結果を利用して、第 1 のストリームを基に第 2 のストリームが生成されるようにしたので、復号処理、符号化処理、および編集処理を繰り返したとしても画質劣化が発生しない。

#### 【図面の簡単な説明】

【図 1】画像の編集を説明する図である。

【図 2】VBV Buffer のオーバーフローを説明する図である。

【図 3】本発明を適用したトランスコーディングシステム 1 の構成を示すブロック図である。

【図 4】図 3 のトランスコーディングシステム 1 のより詳細な構成を示すブロック図である。

【図 5】図 3 の復号装置 1 0 2 に内蔵されるデコーダ 2 5 1 の構成を示すブロック図である。

【図 6】マクロブロックの画素を説明する図である。

【図 7】符号化パラメータが記録される領域を説明する図である。

【図 8】Ancillary Data Packet の例を示す図である。

【図 9】カウンタ値を分割して格納するデータを説明する図である。

【図 1 0】カウンタ値を分割して格納するデータを説明する図である。

【図 1 1】カウンタ値を分割して格納するデータを説明する図である。

【図 1 2】ベースバンドデジタルビデオ信号の LSB に多重化されている符号化パラメータにカウンタ値を多重化するときの、ヒストリ情報多重化装置 1 0 3 およびカウンタ値多重化装置 1 0 5 に対応する機能の構成を説明する図である。

【図 1 3】ベースバンドデジタルビデオ信号の輝度または色差のブランキング部分に多重化されている符号化パラメータにカウンタ値を多重化するときの、ヒストリ情報多重化装置 1 0 3 およびカウンタ値多重化装置 1 0 5 に対応する機能の構成を説明する図である。

【図 1 4】図 3 の符号化装置 1 1 6 に内蔵されるエンコーダ 3 0 1 の構成を示すブロック図である。

【図 1 5】図 3 のトランスコーディングシステム 1 が実際に使用される状態を示す図である。

【図 1 6】密結合されたトランスコーディングシステム 1 の構成を示すブロック図である。

【図 1 7】カウンタ 3 6 2 の構成例を示す図である。

【図 1 8】ビデオシーケンスのストリームのシンタックスを説明する図である。

【図 1 9】図 1 8 のシンタックスの構成を説明する図である。

【図 2 0】固定長の履歴情報を記録する history\_stream () のシンタックスを説明する図である。

【図 2 1】固定長の履歴情報を記録する history\_stream () のシンタックスを説明する図である。

【図 2 2】固定長の履歴情報を記録する history\_stream () のシンタックスを説明する図である。

【図 2 3】固定長の履歴情報を記録する history\_stream () のシンタックスを説明する図である。

【図 2 4】固定長の履歴情報を記録する history\_stream () のシンタックスを説明する図である。

【図 2 5】固定長の履歴情報を記録する history\_stream () のシンタックスを説明する図である。

【図 2 6】固定長の履歴情報を記録する history\_stream () のシンタックスを説明する図である。

【図 2 7】可変長の履歴情報を記録する history\_stream () のシンタックスを説明する図である。

【図 2 8】sequence\_header () のシンタックスを説明する図である。

【図 2 9】sequence\_extension () のシンタックスを説明する図である。

【図 3 0】extension\_and\_user\_data () のシンタックスを説明する図である。

【図 3 1】user\_data () のシンタックスを説明する図である。

【図 3 2】group\_of\_pictures\_header () のシンタックスを説明する図である。

【図 3 3】picture\_header () のシンタックスを説明する図である。

【図 3 4】picture\_coding\_extension () のシンタックスを説明する図である。

【図 3 5】extension\_data () のシンタックスを説明する図である。

【図 3 6】quant\_matrix\_extension () のシンタックスを説明する図である。

【図 3 7】copyright\_extension () のシンタックスを説明する図である。

【図 3 8】picture\_display\_extension () のシンタックスを説明する図である。

【図 3 9】picture\_data () のシンタックスを説明する図である。

【図 4 0】slice () のシンタックスを説明する図である。

【図 4 1】macroblock () のシンタックスを説明する図である。

【図 4 2】macroblock\_modes () のシンタックスを説明する図である。



【図 4 3】 motion\_vectors(s) のシンタックスを説明する図である。

【図 4 4】 motion\_vector(r, s) のシンタックスを説明する図である。

【図 4 5】 I ピクチャに対するmacroblock\_typeの可変長符号を説明する図である。

【図 4 6】 Pピクチャに対するmacroblock\_typeの可変長符号を説明する図である。

【図 4 7】 Bピクチャに対するmacroblock\_typeの可変長符号を説明する図である。

【図４８】本発明を適用したトランスコーディングシステム１の他の構成を示すブロック図である。

【図 4 9】履歴情報の項目の組み合わせを説明する図である。

【図50】図48のトランスコーディングシステム1の動作を説明するフローチャートである。

【図 5 1】 `re_coding_stream_info()` のシンタックスを説明する図である。

【図 5 2】 red\_bw\_flag, red\_bw\_indicatorを説明する図である。

【図 5 3】Re\_Coding Information Bus macroblock for  
ationを説明する図である。

【図 5 4】 Picture rate elementsを説明する図である。

【図 5 5】 Picture rate elementsを説明する図である。

【図 5 6】 Picture rate elementsを説明する図である。

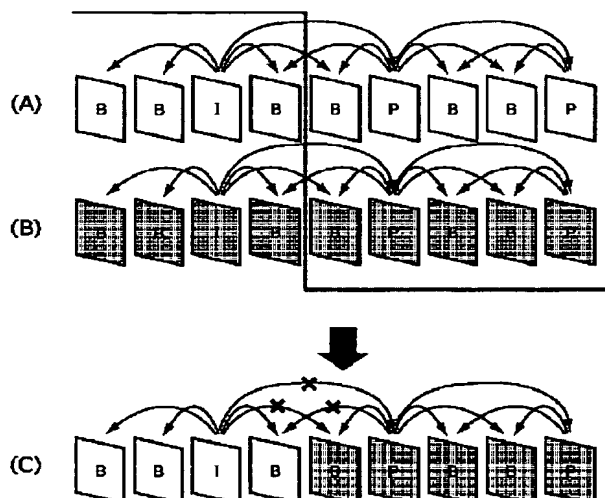
【図 5 7】 Re\_Coding Information Busが記録される領域を説明する図である。

【符号の説明】

1 トランスコーディングシステム, 11 ビデオ復

システム, 12 ビデオ符号化システム, 101  
カウンタ, 102 復号装置, 103 ヒストリ情報  
多重化装置, 104 ヒストリデコーディング装置,  
105 カウンタ値多重化装置, 111 カウンタ値  
分離装置, 112 フォーマット変換装置, 113  
カウンタ, 114 比較装置, 115 ヒストリ  
情報分離装置, 116 符号化装置, 117 ヒス  
トリエンコーディング装置, 201 ユーザデータデ  
コード, 202 コンバータ, 203 ヒストリV  
LD, 211 ヒストリVLC, 212 コンバー  
タ, 213 ユーザデータフォーマッタ, 251  
デコード, 261 受信バッファ, 262 可変長  
復号回路, 263 逆量子化回路, 264 IDCT回  
路, 265 演算器, 266 動き補償回路, 26  
7 フレームメモリ, 271 タイミング信号発生装  
置, 272 カウンタ値フォーマット変換装置, 2  
73 符号化パラメータフォーマット変換装置, 274  
シリアルパラレル変換装置, 275 スイッチ,  
281 タイミング信号発生回路, 282 スイッ  
チ, 301 エンコード, 310 動きベクトル検  
出回路, 311 フレームメモリ, 312 Frame/Field予測モード切り替え回路, 313 演算器, 31  
5 Frame/FieldDCTモード切り替え回路, 316 DC  
T回路, 317 量子化回路, 318 可変長符号化  
回路, 319 送信バッファ, 320 逆量子化回  
路, 321 IDCT回路, 322 演算器, 323  
フレームメモリ, 324 動き補償回路, 330  
コントローラ, 351 SDTI, 361 フォー  
マット変換装置, 362 カウンタ, 363 比較  
装置, 364 カウンタ, 365 フォーマット変  
換装置, 381 カウンタ, 382 AND回路

【图 1】

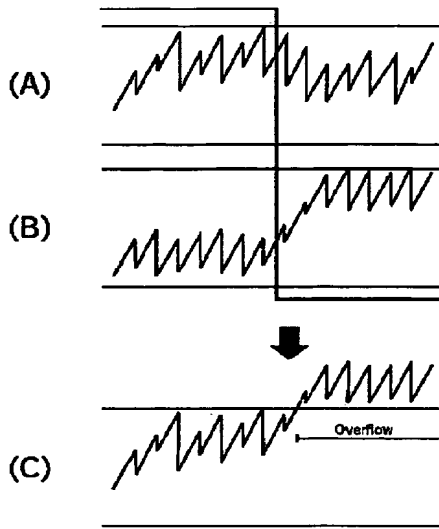


【图 6】

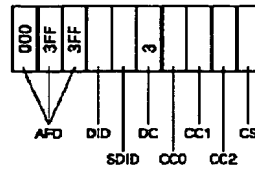
0	1	2	.....	13	14	15
16	17	18	.....	29	30	31
32	33	34	.....	45	46	47
.	.	.		.	.	.
.	.	.		.	.	.
.	.	.		.	.	.
.	.	.		.	.	.
.	.	.		.	.	.
208	209	210	.....	221	222	223
224	225	226	.....	237	238	239
240	241	242	.....	251	254	255

## マクロブロック

【図2】



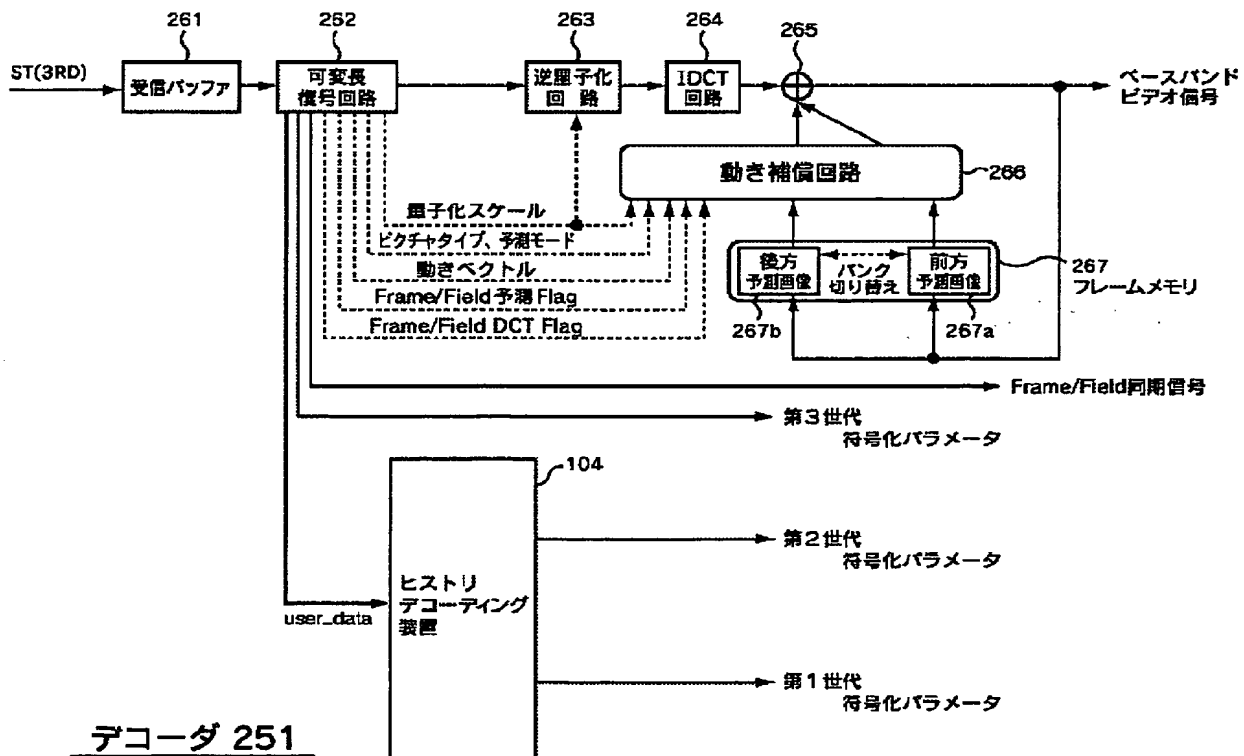
【図8】



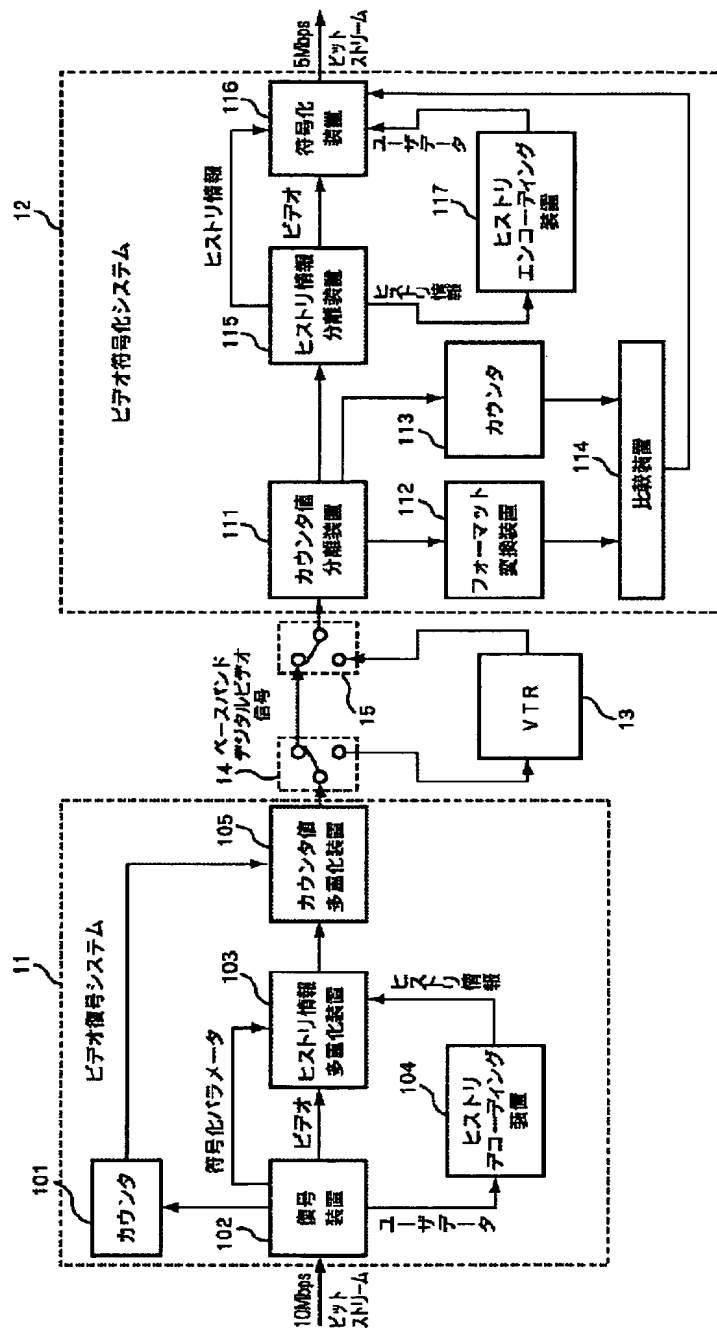
【図9】

CC0(10-bit word)		割り当て
B9		偶数パリティの否定値
B8		偶数パリティ
B7		カウンタ値[5]
B6		カウンタ値[4]
B5		カウンタ値[3]
B4		カウンタ値[2]
B3		カウンタ値[1]
B2		カウンタ値[0]
B1		"0"
B0		"0"

【図5】

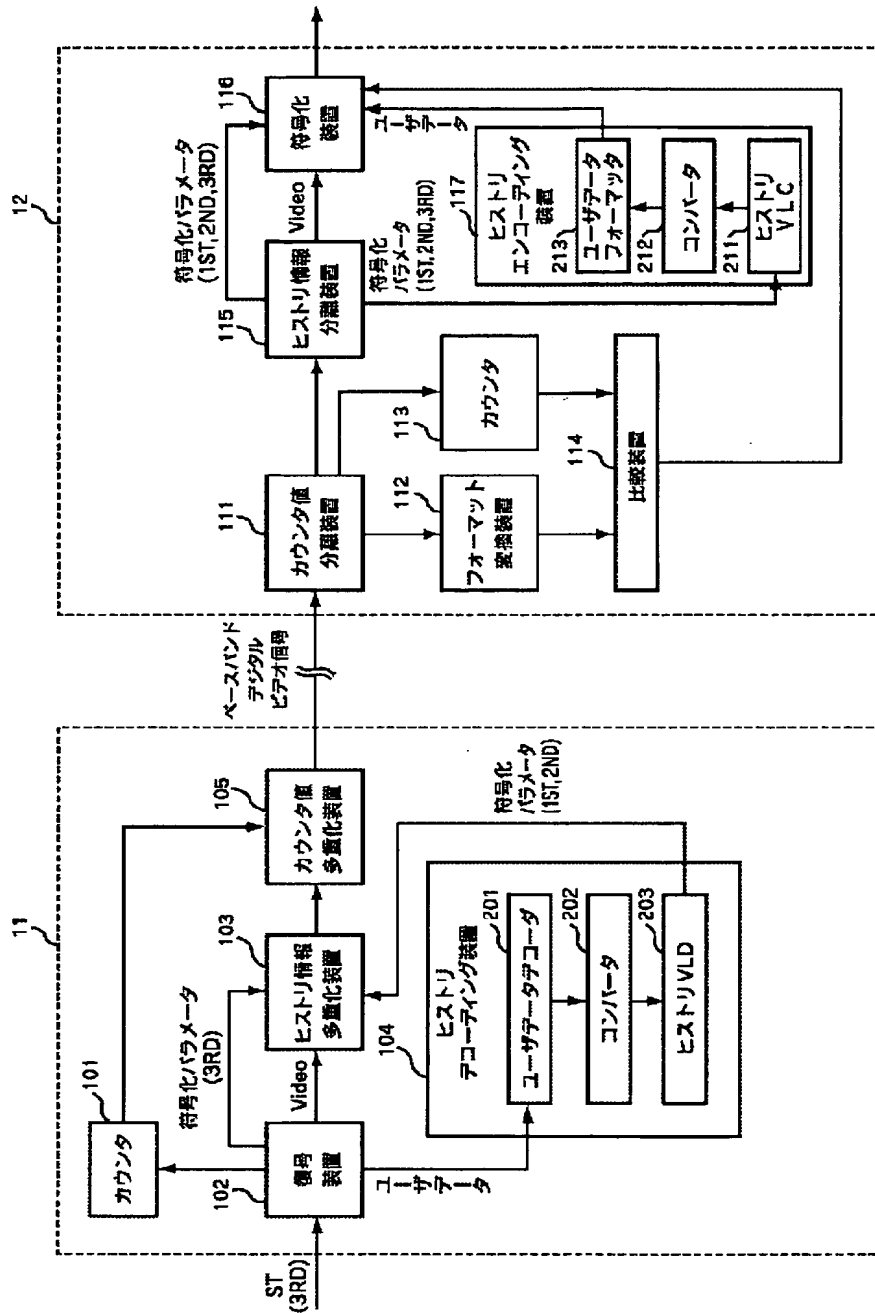


【図3】



トランスコーディングシステム1

【図4】



【図 7】

D9	Cb[0][9]	Y[0][9]	Cr[0][9]	Y[1][9]	Cb[1][9]	Y[2][9]	Cr[1][9]	Y[3][9]	画像データ領域
D8	Cb[0][8]	Y[0][8]	Cr[0][8]	Y[1][8]	Cb[1][8]	Y[2][8]	Cr[1][8]	Y[3][8]	
D7	Cb[0][7]	Y[0][7]	Cr[0][7]	Y[1][7]	Cb[1][7]	Y[2][7]	Cr[1][7]	Y[3][7]	
D6	Cb[0][6]	Y[0][6]	Cr[0][6]	Y[1][6]	Cb[1][6]	Y[2][6]	Cr[1][6]	Y[3][6]	
D5	Cb[0][5]	Y[0][5]	Cr[0][5]	Y[1][5]	Cb[1][5]	Y[2][5]	Cr[1][5]	Y[3][5]	
D4	Cb[0][4]	Y[0][4]	Cr[0][4]	Y[1][4]	Cb[1][4]	Y[2][4]	Cr[1][4]	Y[3][4]	
D3	Cb[0][3]	Y[0][3]	Cr[0][3]	Y[1][3]	Cb[1][3]	Y[2][3]	Cr[1][3]	Y[3][3]	
D2	Cb[0][2]	Y[0][2]	Cr[0][2]	Y[1][2]	Cb[1][2]	Y[2][2]	Cr[1][2]	Y[3][2]	
D1	第 1 世代		第 2 世代		第 3 世代				歴史情報領域
D0									
	Cb[0][x]	Y[0][x]	Cr[0][x]	Y[1][x]	Cb[1][x]	Y[2][x]	Cr[1][x]	Y[3][x]	

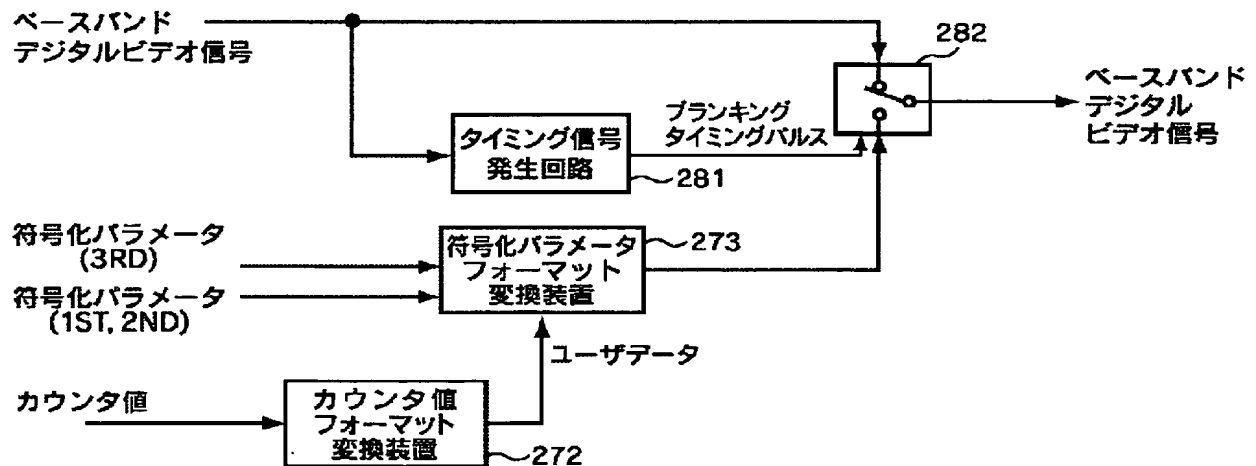
【図 10】

CC1(10-bit word)	割り当て
B9	偶数パリティの否定値
B8	偶数パリティ
B7	カウンタ値[11]
B6	カウンタ値[10]
B5	カウンタ値[9]
B4	カウンタ値[8]
B3	カウンタ値[7]
B2	カウンタ値[6]
B1	"0"
B0	"0"

【図 11】

CC2(10-bit word)	割り当て
B9	偶数パリティの否定値
B8	偶数パリティ
B7	"0"
B6	"0"
B5	カウンタ値[15]
B4	カウンタ値[14]
B3	カウンタ値[13]
B2	カウンタ値[12]
B1	"0"
B0	"0"

【図 13】

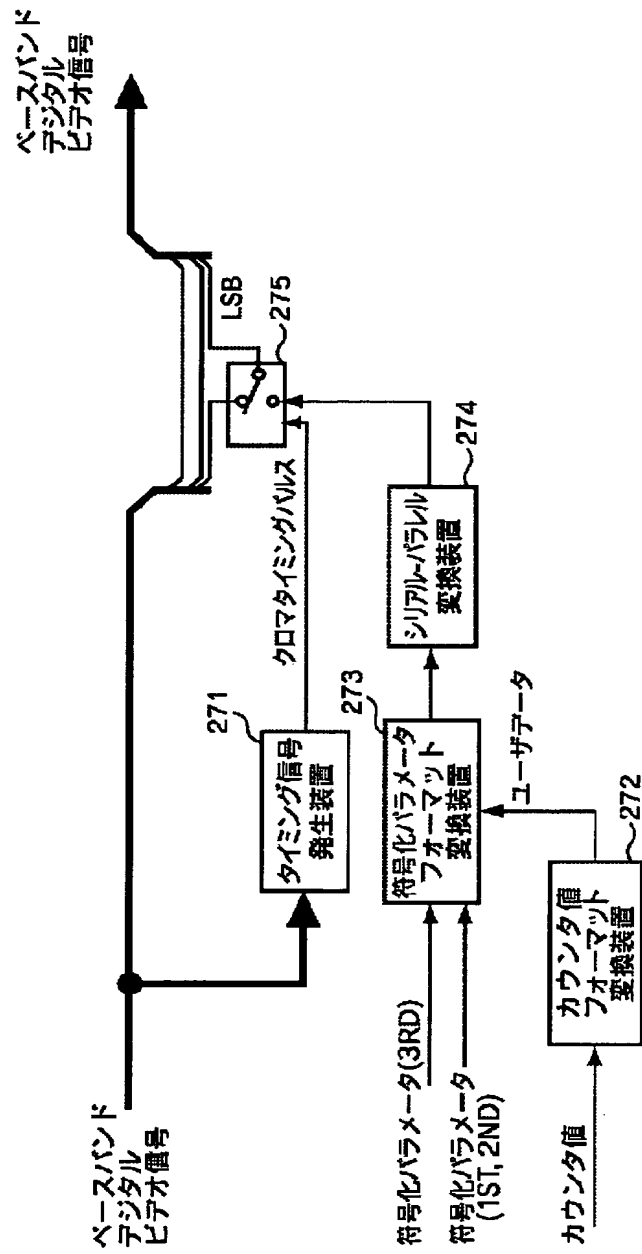


【図 26】

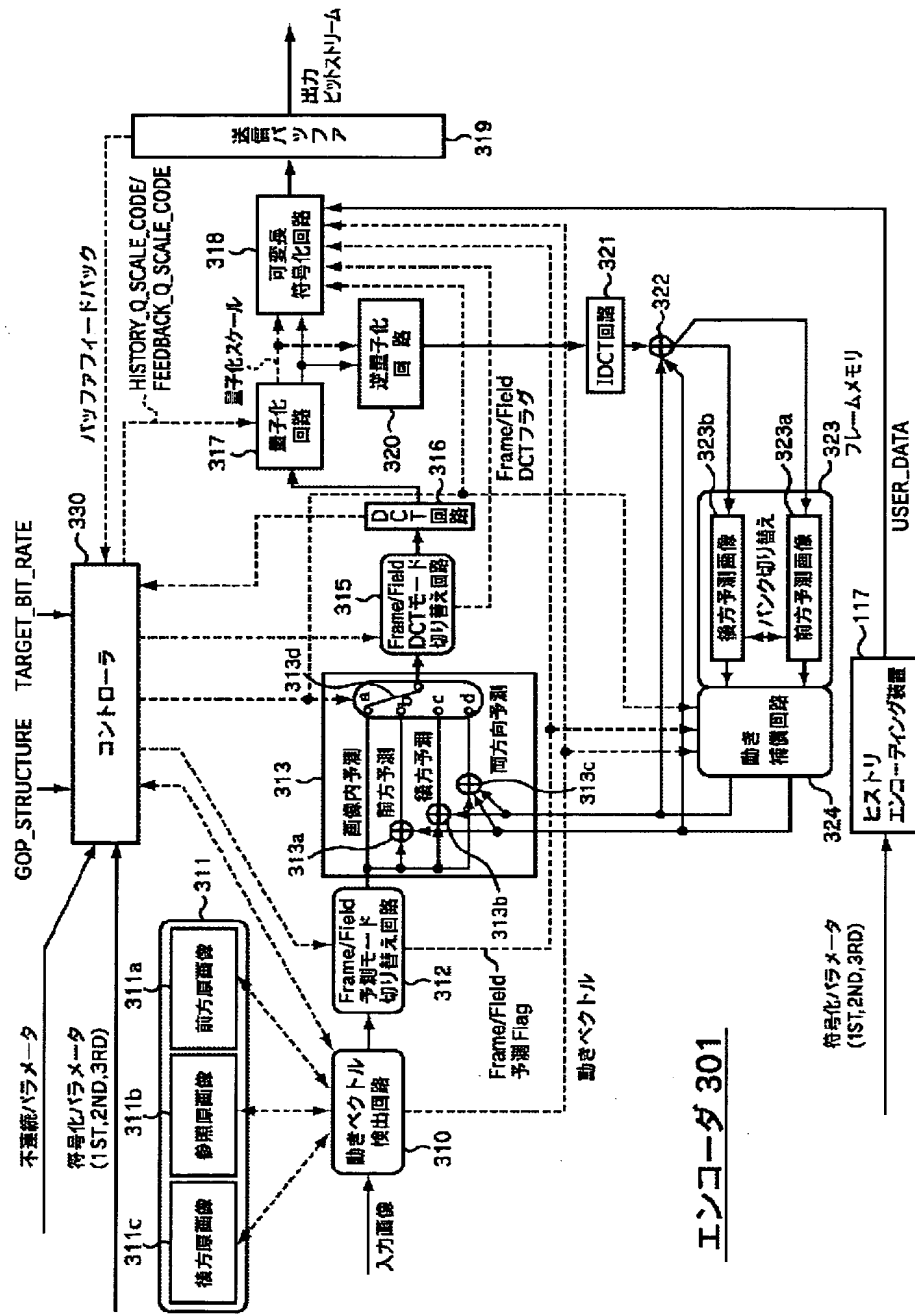
history stream(20-7)

	bits	value
num_other_bits	7	
marker_bit	1	1
}		

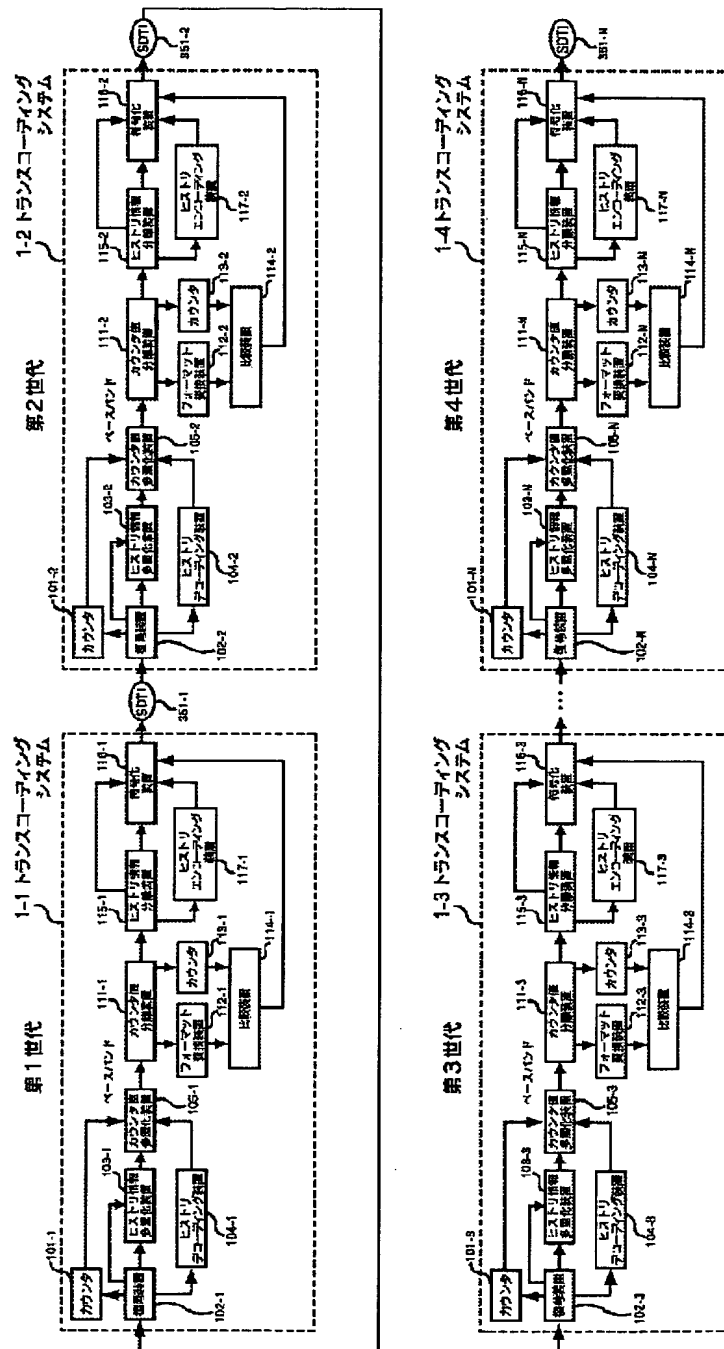
【図12】



【図14】

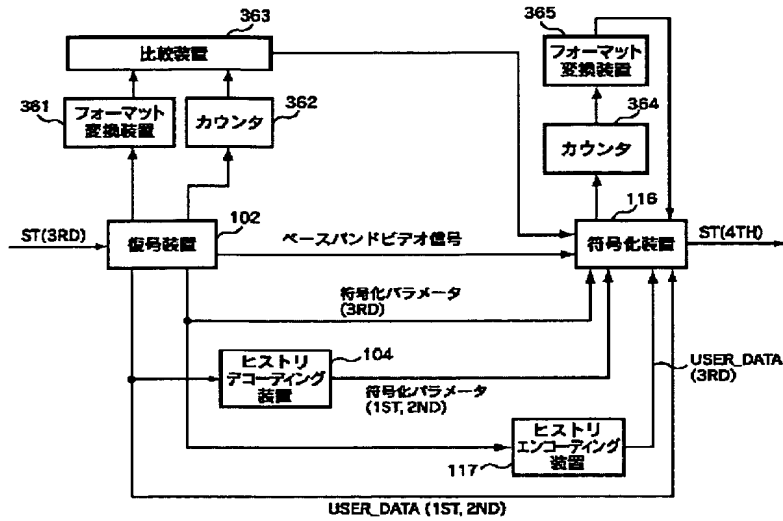


【図15】

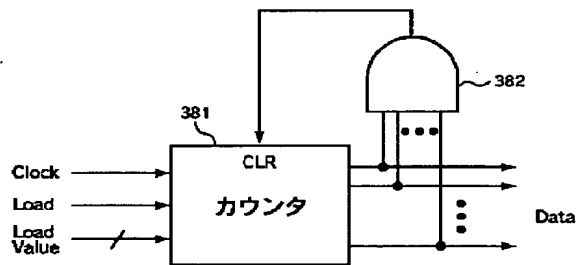




【図 16】



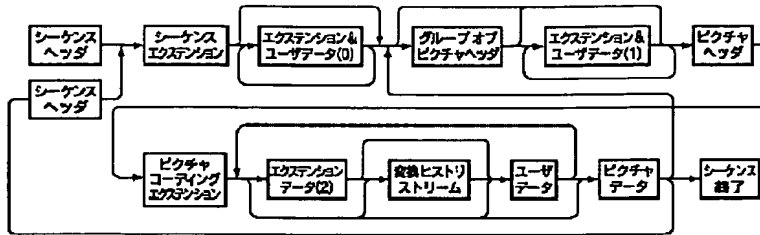
【図 17】



【図 18】

stream with history data		
	No. of bits	Mnemonic
video_sequence ( )		
next_start_code ( )		
sequence_header ( )		
sequence_extension ( )		
do		
extension_and_user_data (0)		
do		
if(nextbits() == group_start_code){		
group_of_pictures_header(1)		
extension_and_user_data (1)		
}		
picture_header()		
picture_coding_extensions()		
while((nextbits() != extension_start_code)		
(nextbits() != user_data_start_code)){		
if(nextbits() != extension_start_code){		
extension_data(2)		
if(nextbits() != user_data_start_code){		
user_data_start_code	32	bslbf
video_continuity_counter_ID	16	bslbf
marker_bit_1	1	bslbf
video_continuity_counter	16	uimsbf
marker_bit_2	1	bslbf
video_continuity_counter_CRC	16	bslbf
if(nextbits() != History_Data_ID){		
History_Data_ID	8	bslbf
converted_history_stream()		
else{		
user_data()		
}		
picture_data()		
while((nextbits() != picture_start_code)		
(nextbits() != groupe_start_code)){		
if(nextbits() != sequence_end_code){		
sequence_header ( )		
sequence_extension ( )		
}		
while(nextbits() != sequence_end_code)		
sequence_end_code	32	bslbf

【図 19】



【図 20】

History stream(20-1)			
history_stream( ) {		bits	value
sequence_header			
sequence_header_code	32	000001B3	
sequence_header_present_flag	1		
horizontal_size_value	12		
marker_bit	1	1	
vertical_size_value	12		
aspect_ratio_information	4		
frame_rate_code	4		
marker_bit	1	1	
bit_rate_value	18		
marker_bit	1	1	
vbv_buffer_size_value	10		
constrained_parameter_flag	1	0	
load_intra_quantiser_matrix	1		
load_non_intra_quantiser_matrix	1		
marker_bits	5	1F	
intra_quantiser_matrix[64]	8*64		
non_intra_quantiser_matrix[64]	8*64		
sequence_extension			
extension_start_code	32	000001B5	
extension_start_code_identifier	4	1	
sequence_extension_present_flag	1		
profile_and_level_indication	8		
progressive_sequence	1		
chroma_format	2		
horizontal_size_extension	2		
vertical_size_extension	2		
marker_bit	1	1	
bit_rate_extension	12		
vbv_buffer_size_extension	8		
low_delay	1		
marker_bit	1	1	

【図 27】

history_stream( ) {		
next_start_code( )	No. of bits	Mnemonic
sequence_header( )		
sequence_extension( )		
extension_and_user_data(0)		
if(nextbits( ) == group_start_code){		
group_of_pictures_header( )		
extension_and_user_data(1)		
}		
picture_header( )		
picture_coding_extension( )		
re_coding_stream_info( )		
extensions_and_user_data(2)		
picture_data( )		
sequence_end_code	32	bslbf
}		

【図 21】

History stream(20-2)			
		bits	value
frame_rate_extension_n		2	
frame_rate_extension_d		5	
marker_bits		6	3F
sequence_display_extension			
extension_start_code	32	000001B5	
extension_start_code_identifier	4	2	
sequence_display_extension_present_flag	1		
video_format	3		
colour_description	1		
colour_primaries	8		
transfer_characteristics	8		
marker_bit	1	1	
matrix_coefficients	8		
display_horizontal_size	14		
marker_bit	1	1	
display_vertical_size	14		
marker_bit	1	1	
macroblock_assignment_in_user_data			
macroblock_assignment_present_flag	1		
marker_bit	7	7F	
v_phase	8		
h_phase	8		
group_of_picture_header			
group_start_code	32	000001B8	
group_of_picture_header_present_flag	1		
time_code	25		
closed_gop	1		
broken_link	1		
marker_bits	4	F	
picture_header			
picture_start_code	32	00000100	

【図 30】

extension_and_user_data(1){		
while((nextbits( ) == extension_start_code)	No. of bits	Mnemonic
(nextbits( ) == user_data_start_code)){		
if((i==2) && (nextbits( ) == extension_start_code)){		
extension_data( )		
if(nextbits( ) == user_data_start_code)		
user_data( )		
}		
}		

【図 22】

History stream(20-3)		
	bits	value
temporal_reference	10	
picture_coding_type	3	
marker_bit	1	1
vbv_delay	16	
full_pel_forward_vector	1	
forward_f_code	3	
full_pel_backward_vector	1	
marker_bit	1	1
backward_f_code	3	
marker_bit	1	1
picture_coding_extension		
extension_start_code	32	00000185
extension_start_code_identifier	4	8
f_code[0][0]	4	
f_code[0][1]	4	
f_code[1][0]	4	
f_code[1][1]	4	
intra_dc_precision	2	
picture_structure	2	
top_field_first	1	
frame_pred_frame_dct	1	
concealment_motion_vectors	1	
q_scale_type	1	
marker_bit	1	1
intra_vlc_format	1	
alternate_scan	1	
repeat_first_field	1	
chroma_420_type	1	
progressive_frame	1	
composite_display_flag	1	
v_axis	1	
field_sequence	3	
sub_carrier	1	
burst_amplitude	7	

【図 28】

sequence_header( )		
	No. of bits	Mnemonic
sequence_header_code( )	32	bslbf
horizontal_size_value	12	ulmsbf
vertical_size_value	12	ulmsbf
aspect_ratio_information	4	ulmsbf
frame_rate_code	4	ulmsbf
bit_rate_value	18	ulmsbf
marker_bit	1	bslbf
vbv_buffer_size_value	10	ulmsbf
constrained_parameters_flag	1	bslbf
load_intra_quantiser_matrix	1	ulmsbf
if(load_intra_quantiser_matrix)		
intra_quantiser_matrix[64]	8*64	ulmsbf
load_non_intra_quantiser_matrix	1	ulmsbf
if(load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix[64]	8*64	ulmsbf
next_start_code( )		
}		

【図 31】

user_data( )		
	No. of bits	Mnemonic
user_data_start_code	32	bslbf
while(nextbits( ) != '0000 0000 0000 0000 0001') {		
user_data	8	ulmsbf
}		
next_start_code( )		
}		

【図 23】

History stream(20-4)		
	bits	value
marker_bits	1	1
sub_carrier_phase	8	
quant_matrix_extension		
extension_start_code	32	00000185
extension_start_code_identifier	4	3
quant_matrix_extension_present_flag	1	
load_intra_quantiser_matrix	1	
marker_bits	2	3
intra_quantiser_matrix[64]	8*64	
load_non_intra_quantiser_matrix	1	
marker_bits	7	7F
non_intra_quantiser_matrix[64]	8*64	
load_chroma_intra_quantiser_matrix	1	
marker_bits	7	7F
chroma_intra_quantiser_matrix[64]	8*64	
load_chroma_non_intra_quantiser_matrix	1	
marker_bits	7	7F
chroma_non_intra_quantiser_matrix[64]	8*64	
copyright_extension		
extension_start_code	32	00000185
extension_start_code_identifier	4	4
copyright_extension_present_flag	1	
copyright_flag	1	
copyright_identifier	8	
original_or_copy	1	
marker_bit	1	
copyright_number_1	20	
marker_bit	1	
copyright_number_2	22	
marker_bit	1	
copyright_number_3	22	3F
marker_bits	6	

【図 29】

sequence_extension( )		
	No. of bits	Mnemonic
extension_start_code	32	bslbf
extension_start_code_identifier	4	ulmsbf
profile_and_level_indication	8	ulmsbf
progressive_sequence	1	ulmsbf
chroma_format	2	ulmsbf
horizontal_size_extension	2	ulmsbf
vertical_size_extension	2	ulmsbf
bit_rate_extension	12	ulmsbf
marker_bit	1	bslbf
vbv_buffer_size_extension	8	ulmsbf
low_delay	1	ulmsbf
frame_rate_extension_n	2	ulmsbf
frame_rate_extension_d	5	ulmsbf
next_start_code( )		
}		

【図 32】

group_of_pictures_header( )		
	No. of bits	Mnemonic
group_start_code	32	bslbf
time_code	25	bslbf
closed_gap	1	ulmsbf
broken_link	1	ulmsbf
next_start_code( )		
}		

【図 2 4】

History stream(20-5)		
	bits	value
picture_display_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	7
picture_display_extension_present_flag	1	
frame_centre_horizontal_offset_1	16	
marker_bit	1	1
frame_centre_vertical_offset_1	16	
marker_bit	1	1
frame_centre_horizontal_offset_2	16	
marker_bit	1	1
frame_centre_vertical_offset_2	16	
marker_bit	1	1
frame_centre_horizontal_offset_3	16	
marker_bit	1	1
frame_centre_vertical_offset_3	16	
marker_bit	6	3F
re_coding_stream_information		
user_data_start_code	32	000001B2
re_coding_stream_info_ID	16	91EC
red_bw_flag	1	
red_bw_indicator	2	
marker_bit	5	1F
user_data		
user_data_start_code	32	000001B2
user_data	2049	
while(macroblock != macroblock_count){		
macroblock		
macroblock_address_h	8	
macroblock_address_v	8	
slice_header_present_flag	1	
skipped_macroblock_flag	1	
marker_bit	1	1
macroblock_modes()		
macroblock_quant	1	
macroblock_motion_forward	1	
macroblock_motion_backward	1	
macroblock_pattern	1	
macroblock_intra	1	

【図 3 3】

picture_header()		
	No. of bits	Mnemonic
picture_start_code	32	bslbf
temporal_reference	10	uimbsf
picture_coding_type	3	uimbsf
vbv_delay	16	uimbsf
if(picture_coding_type==2    picture_coding_type==3){		
full_pel_forward_vector	1	bslbf
forward_f_code	3	bslbf
}		
if(picture_coding_type==3){		
full_pel_forward_vector	1	bslbf
backward_f_code	3	bslbf
}		
while(nextbits()=='1'){		
extra_bit_picture/*with the value '1'*/	1	uimbsf
extra_information_picture	8	uimbsf
}		
extra_bit_picture/*with the value '0'*/	1	uimbsf
next_start_code()		
}		

【図 3 9】

picture_data()		
	No. of bits	Mnemonic
while(nextbits() == slice_start_code){		
slice()		
}		
next_start_code()		
}		

【図 2 5】

History stream(20-6)		
	bits	value
spatial_temporal_weight_code_flag	1	
frame_motion_type	2	
field_motion_type	2	
dct_type	1	
marker_bits	2	3
quantiser_scale_code	5	
marker_bits	3	7
PMV[0][0][0]	14	
marker_bits	2	3
PMV[0][0][1]	14	
motion_vertical_field_select[0][0]	1	
marker_bit	1	1
PMV[0][1][0]	14	
marker_bits	2	3
PMV[0][1][1]	14	
motion_vertical_field_select[0][1]	1	
marker_bit	1	1
PMV[1][0][0]	14	
marker_bits	2	3
PMV[1][0][1]	14	
motion_vertical_field_select[1][0]	1	
marker_bit	1	1
PMV[1][1][0]	14	
marker_bits	2	3
PMV[1][1][1]	14	
motion_vertical_field_select[1][1]	1	
marker_bit	1	1
coded_block_pattern	12	
marker_bits	4	F
num_mv_bits	8	
num_coef_bits	14	
marker_bits	2	3

【図 3 4】

picture_coding_extension()		
	No. of bits	Mnemonic
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimbsf
f_code [0] [0] /*forward horizontal*/	4	uimbsf
f_code [0] [1] /*forward vertical*/	4	uimbsf
f_code [1] [0] /*backward horizontal*/	4	uimbsf
f_code [1] [1] /*backward vertical*/	4	uimbsf
intra_dc_precision	2	uimbsf
picture_structure	2	uimbsf
top_field_first	1	uimbsf
frame_pred_frame_dct	1	uimbsf
concealment_motion_vectors	1	uimbsf
q_scale_type	1	uimbsf
intra_vlc_format	1	uimbsf
alternate_scan	1	uimbsf
repeat_first_field	1	uimbsf
chroma_420_type	1	uimbsf
progressive_frame	1	uimbsf
composite_display_flag	1	uimbsf
if(composite_display_flag){		
v_axis	1	uimbsf
field_sequence	3	uimbsf
sub_carrier	1	uimbsf
burst_amplitude	7	uimbsf
sub_carrier_phase	8	uimbsf
}		
next_start_code()		
}		

【図 3 5】

extension_data( ) {	No. of bits	Mnemonic
while(nextbits( )==extension_start_code) {		
extension_start_code	32	bslbf
if (nextbits( )=="Quant Matrix Extension ID")		
quant_matrix_extension( )		
else if (nextbits( )=="Copyright Extension ID")		
copyright_extension( )		
else		
picture_display_extension( )		
}		

【図 3 7】

copyright_extension( ) {	No. of bits	Mnemonic
extension_start_code_identifier	4	uimbsf
copyright_flag	1	bslbf
copyright_identifier	8	uimbsf
original_or_copy	1	bslbf
reserved	7	uimbsf
marker_bit	1	bslbf
copyright_number_1	20	uimbsf
marker_bit	1	bslbf
copyright_number_2	22	uimbsf
marker_bit	1	bslbf
copyright_number_3	22	uimbsf
next_start_code( )		

【図 4 0】

slice( ) {	No. of bits	Mnemonic
slice_start_code	32	bslbf
slice_quantiser_scale_code	5	uimbsf
if (nextbit( )=='1') {		
intra_slice_flag	1	bslbf
intra_slice	1	uimbsf
reserved_bits	7	uimbsf
while(nextbits( )=='1') {		
extra_bit_slice/*with the value '1'*/	1	uimbsf
extra_information_slice	8	uimbsf
}		
extra_bit_slice/*with the value '0'*/	1	uimbsf
do {		
macroblock( )		
} while(nextbit( )!='000 0000 0000 0000 0000')		
next_start_code( )		

【図 4 3】

motion_vectors(s) {	No. of bits	Mnemonic
if (motion_vector_count==1) {		
if ((mv_format==field)&&(dmv!=1))		
motion_vertical_field_select [0][s]	1	uimbsf
motion_vector(0,s)		
} else {		
motion_vertical_field_select [0][s]	1	uimbsf
motion_vector(0,s)		
motion_vertical_field_select [1][s]	1	uimbsf
motion_vector(1,s)		
}		

【図 3 6】

quant_matrix_extension( ) {	No. of bits	Mnemonic
extension_start_code_identifier	4	uimbsf
load_intra_quantiser_matrix	1	uimbsf
if (load_intra_quantiser_matrix)		
intra_quantiser_matrix [64]	8*64	uimbsf
load_non_intra_quantiser_matrix	1	uimbsf
if (load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix [64]	8*64	uimbsf
load_chroma_intra_quantiser_matrix	1	uimbsf
if (load_chroma_intra_quantiser_matrix)		
chroma_intra_quantiser_matrix [64]	8*64	uimbsf
load_chroma_non_intra_quantiser_matrix	1	uimbsf
if (load_chroma_non_intra_quantiser_matrix)		
chroma_non_intra_quantiser_matrix [64]	8*64	uimbsf
next_start_code( )		

【図 3 8】

picture_display_extension( ) {	No. of bits	Mnemonic
extension_start_code_identifier	4	uimbsf
for(i=0;i:number_of_frame_centre_offsets;i++) {		
frame_centre_horizontal_offset	16	simbsf
marker_bit	1	bslbf
frame_centre_vertical_offset	16	simbsf
marker_bit	1	bslbf
}		
next_start_code( )		

【図 4 1】

macroblock( ) {	No. of bits	Mnemonic
while(nextbits( )=="0000 0001 000")		
macroblock_escape	11	bslbf
macroblock_address_increment	1-11	viclbf
macroblock_modes( )		
if (macroblock_quant)		
macroblock_quantiser_scale_code	5	uimbsf
if (macroblock_motion_forward		
(macroblock_intra && concealment_motion_vectors))		
motion_vectors(0)		
if (macroblock_motion_backward)		
motion_vectors(1)		
if (macroblock_intra && concealment_motion_vectors)		
marker_bit	1	bslbf
}		

【図 4 4】

motion_vectors(r,s) {	No. of bits	Mnemonic
motion_code[r][s][0]	1-11	viclbf
if ((f_code[s][0]!=1)&&(motion_code[r][s][0]!=0))		
motion_residual [r][s][0]	1-8	uimbsf
if (dmv==1)		
dmvector [0]	1-2	viclbf
motion_code[r][s][1]	1-11	viclbf
if ((f_code[s][1]!=1)&&(motion_code[r][s][1]!=0))		
motion_residual [r][s][1]	1-8	uimbsf
if (dmv==1)		
dmvector [1]	1-2	viclbf

【図 4 2】

macroblock_modes( ) {	No. of bits	Mnemonic
macroblock_type	1-9	vclbf
if (macroblock_motion_forward		
macroblock_motion_backward) {		
if (picture_structure=='frame') {		
if (frame_pred_frame_dct==0)		
frame_motion_type	2	uimsbf
} else {		
field_motion_type	2	uimsbf
}		
if ((picture_structure=="Frame picture") &&		
(frame_pred_frame_dct==0) &&		
(dct_type_flag==1)) {		
dct_type	1	uimsbf
}		

【図 4 6】

macroblock_type VLC code	
	Description
1	MC, Coded
01	No MC, Coded
001	MC, Not Coded
0001 1	Intra
0001 0	MC, Coded, Quant
0000 1	No MC, Coded, Quant
0000 01	Intra, Quant

【図 4 5】

macroblock_type VLC code	
	Description
1	Intra
01	Intra, Quant

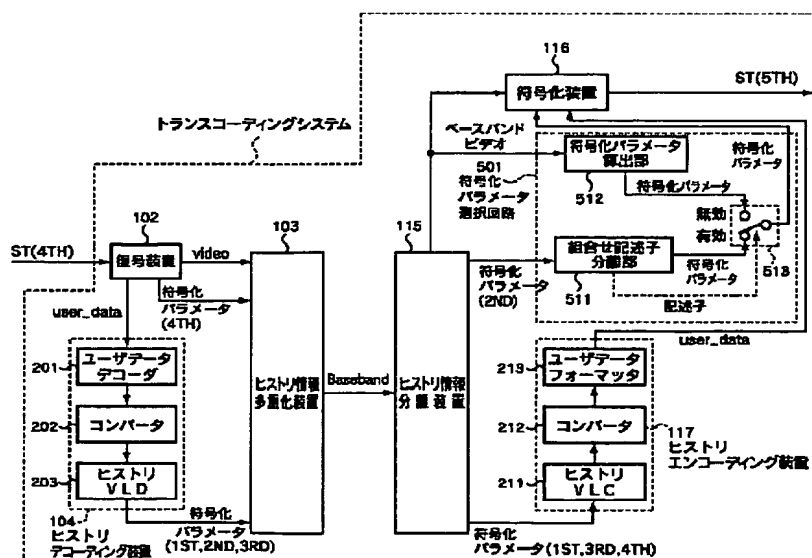
【図 5 2】

項目の組合せ	red_bw_flag	
	red_bw_flag	red_bw_indicator
組合せ 1	0	—
組合せ 2	1	0
組合せ 3	1	1
組合せ 4	1	2
組合せ 5	1	3

【図 4 7】

macroblock_type VLC code	
	Description
10	Interp, Not Coded
11	Interp, Coded
010	Bwd, Not Coded
011	Bwd, Coded
0010	Fwd, Not Coded
0011	Fwd, Coded
0001 1	Intra
0001 0	Interp, Coded, Quant
0000 11	Fwd, Coded, Quant
0000 10	Bwd, Coded, Quant
0000 01	Intra, Quant

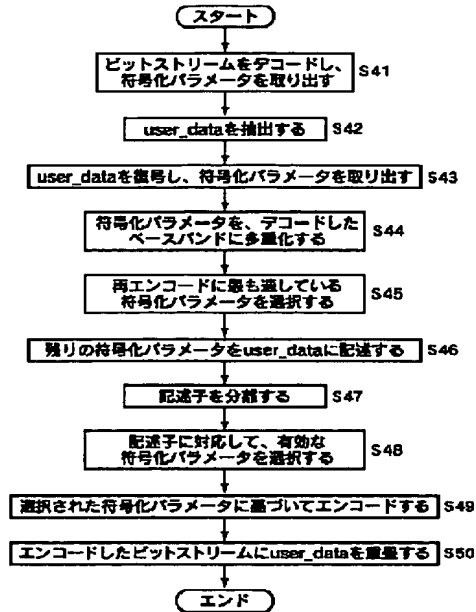
【図 4 8】



【図 5 0】

[illegible]

履歴情報の項目の組合せ



【図 5 1】

【例 5 3】

re_coding_stream_info ( ) {	No. of bits	Mnemonic
user_data_start_code	32	bslbf
re_coding_stream_info_ID	16	bslbf
red_bw_flag	1	ulmsbf
if (red_bw_flag)		
red_bw_indicator	2	ulmsbf
if (!red_bw_flag) {		
for(i=0; i<number_of_macroblock++;) {		
marker_bit	3	bslbf
num_other_bits	7	ulmsbf
num_mv_bits	8	ulmsbf
num_coef_bits	14	ulmsbf
}		
}		
next_start_code( )		
}		

Frame Coding	Line	Field Coding	Reserved															
0	0	0	SRIB_sync_code[11:12]															
			rotating SRIB_mb_ref[15:0]															
1	2	2	picrate_element [picrate_element_index][31:16]															
	3	3	picrate_element [picrate_element_index][15:0]															
2	4	4	mb_quant_idx[2:3]	mb_quant_idx[4:5]	mb_quant_idx[6:7]	mv_mv[0][0][0][12:0]										mv_mv[0][0][0][12:0]		
	5	5	mb_quant_idx[8:9]	mb_quant_idx[10:11]	mb_quant_idx[12:13]	mv_mv[0][0][1][8:0]										mv_mv[0][0][1][8:0]		
3	6	6	mb_mv_idx[0:1]	mb_mv_idx[2:3]	mb_mv_idx[4:5]	mv_mv[0][1][0][12:0]										mv_mv[0][1][0][12:0]		
	7	7	mb_mv_idx[6:7]	mb_mv_idx[8:9]	mb_mv_idx[10:11]	mv_mv[0][1][1][8:0]										mv_mv[0][1][1][8:0]		
4	8	8	Reserved															
	9	9	Reserved															
5	10	10	Reserved															
	11	11	Reserved															
6	12	12	Reserved															
	13	13	Reserved															
7	14	14	Reserved															
	15	15	Reserved															

【图 5 6】

picture rate elements(54-3)

frame_centre_vertical_offset_2	16bit unsigned	16	2393	2408	2	
frame_centre_horizontal_offset_3	16bit unsigned	16	2409	2424	2	
frame_centre_vertical_offset_3	16bit unsigned	16	2425	2440	2	
copyright extension						
copyright flag	1bit flag	1	2441	2441	2	
copyright identifier	8bit code	8	2442	2449	2	
original or copy	1bit flag	1	2450	2450	2	
copyright number	64bit unsigned	64	2451	2514	2	
PTS/DTS						
PTS DTS flag	2bit flag	2	2515	2516	1	
PTS Value	33bit unsigned	33	2517	2549	2	
DTS Value	33bit unsigned	33	2550	2582	2	
spare reserved bits						
spare	41bit unsigned	41	2583	2623		
user data area						
user_data			1684	2624	4287	2
picture rate Information CRC						
32-bit protection CRT	32bit intech	32	4288	4319		

【図54】

picture rate elements(54-1)

parameter	number format	number of bits	bit offset from	bit offset to	flag code [0]	details
MPEG standard flag	1bit flag	1	0	0	3	1->MPEG 1&2-MPEG2
red_bw_flag	1bit flag	1	1	1	3	Default="0"
red_bw_indicator	2bit ul	3	2	4	3	Default="000"
header present flags	2bit flags	2	5	6	3	sequence header present flag, GOP header present flag
Extension start code flags	16 flags	16	7	22	3	indicates if a given extension start code exists. The 16 flags correspond to the 16 codes in table 8.3 of the ISO/IEC 13818-2. Flags placed in the order they are listed.
Other start codes	3 flags	3	23	25	3	user data start code, sequence_end_code, sequence_end_code
sequence header						
horizontal size	14bit ulmsbf	14	26	39	2	includes extension
vertical size	14bit ulmsbf	14	40	53	2	includes extension
aspect_ratio_information	4bit ulmsbf	4	54	57	2	
frame_rate_code	4bit ulmsbf	4	58	61	2	
bit_rate	30bit ulmsbf	30	62	91	2	includes extension, could also be placed in extension
vbv_buffer_size	18bit ulmsbf	18	92	109	2	includes extension
constrained_parameters_flag	1bit flag	1	110	110	2	
sequence extension						
profile_and_level_indication	8bit ulmsbf	8	111	118	2	
progressive_sequence	1bit flag	1	119	119	2	
chroma_format	2bit ulmsbf	2	120	121	2	
low_delay	1bit flag	1	122	122	2	
sequence display extension						
video format	3bit ulmsbf	3	123	125	2	
colour_description	1bit flag	1	126	126	2	
colour primaries	8bit ulmsbf	8	127	134	2	
transfer characteristics	8bit ulmsbf	8	135	142	2	
matrix coefficients	8bit ulmsbf	8	143	150	2	
display_horizontal_size	14bit ulmsbf	14	151	164	2	
display_vertical_size	14bit ulmsbf	14	165	178	2	
group of pictures header						
time code	25bit flag	25	179	203	2	
closed_gop	1bit flag	1	204	204	2	

【図57】

D8	Cb[0][9]	Y[0][9]	Cr[0][9]	Y[1][9]	Cb[1][9]	Y[2][9]	Cr[1][9]	Y[3][9]
D8	Cb[0][8]	Y[0][8]	Cr[0][8]	Y[1][8]	Cb[1][8]	Y[2][8]	Cr[1][8]	Y[3][8]
D7	Cb[0][7]	Y[0][7]	Cr[0][7]	Y[1][7]	Cb[1][7]	Y[2][7]	Cr[1][7]	Y[3][7]
D7	Cb[0][6]	Y[0][6]	Cr[0][6]	Y[1][6]	Cb[1][6]	Y[2][6]	Cr[1][6]	Y[3][6]
D5	Cb[0][5]	Y[0][5]	Cr[0][5]	Y[1][5]	Cb[1][5]	Y[2][5]	Cr[1][5]	Y[3][5]
D4	Cb[0][4]	Y[0][4]	Cr[0][4]	Y[1][4]	Cb[1][4]	Y[2][4]	Cr[1][4]	Y[3][4]
D3	Cb[0][3]	Y[0][3]	Cr[0][3]	Y[1][3]	Cb[1][3]	Y[2][3]	Cr[1][3]	Y[3][3]
D2	Cb[0][2]	Y[0][2]	Cr[0][2]	Y[1][2]	Cb[1][2]	Y[2][2]	Cr[1][2]	Y[3][2]
D1	Cb[0][1]	Y[0][1]	Cr[0][1]	Y[1][1]	Cb[1][1]	Y[2][1]	Cr[1][1]	Y[3][1]
D0	Embedded Aligned MPEG-2 Re-Coding Information Bus	Y[0][0]	Embedded Aligned MPEG-2 Re-Coding Information Bus	Y[1][0]	Embedded Aligned MPEG-2 Re-Coding Information Bus	Y[2][0]	Embedded Aligned MPEG-2 Re-Coding Information Bus	Y[3][0]

【図55】

picture rate elements(54-2)

broken link	1bit flag	1	205	205	2	
picture header						
temporal reference	10bit ulmsbf	10	206	215	1	
picture coding type	3bit ulmsbf	3	216	218	1	
vbv_delay	16bit ulmsbf	16	219	234	1	should be calculated if not present in all cases
full_pel_forward_vector	1bit flag	1	235	235	1	
forward_f_code	3bit ulmsbf	3	236	238	1	
full_pel_backward_vector	1bit flag	1	239	239	1	
backward_f_code	3bit ulmsbf	3	240	242	1	
picture coding extension						
forward_horizontal_f_code	4bit ulmsbf	4	243	246	1	
forward_vertical_f_code	4bit ulmsbf	4	247	250	1	
backward_horizontal_f_code	4bit ulmsbf	4	251	254	1	
backward_vertical_f_code	4bit ulmsbf	4	255	258	1	
intra_dc_precision	2bit ulmsbf	2	259	260	1	
picture structure	2bit ulmsbf	2	261	262	1	
top_field_first	1bit flag	1	263	263	1	
frame_pred_frame_dct	1bit flag	1	264	264	1	
concealment_motion_vectors	1bit flag	1	265	265	1	
q_scale_type	1bit flag	1	266	266	1	
intra_vic_format	1bit flag	1	267	267	1	
alternate_scan	1bit flag	1	268	268	1	
repeat_first_field	1bit flag	1	269	269	1	
chroma_420_type	1bit flag	1	270	270	1	
progressive_frame	1bit flag	1	271	271	1	
composite_display_flag	1bit flag	1	272	272	1	
v_axis	1bit flag	1	273	273	1	
field sequence	3bit ulmsbf	3	274	276	1	
sub_carrier	1bit flag	1	277	277	1	
burst_amplitude	7bit ulmsbf	7	278	284	1	
sub_carrier_phase	8bit ulmsbf	8	285	292	1	
quant matrix extension						
load_intra_quantiser_matrix	1bit flag	1	293	293	1 (1)	
load_non_intra_quantiser_matrix	1bit flag	1	294	294	1 (1)	
load_chroma_intra_quantiser_matrix	1bit flag	1	295	295	1 (1)	
load_chroma_non_intra_quantiser_matrix	1bit flag	1	296	296	1 (1)	
intra_quantiser_matrix[64]	64*0..255	512	297	808	2 (1)	
non_intra_quantiser_matrix[64]	64*0..255	512	809	1320	2 (1)	
chroma_intra_quantiser_matrix[64]	64*0..255	512	1321	1832	2 (1)	
chroma_non_intra_quantiser_matrix[64]	64*0..255	512	1833	2344	2 (1)	
picture display extension						
frame_centre_horizontal_offset_1	16bit ulmsbf	16	2345	2360	2	
frame_centre_vertical_offset_1	16bit ulmsbf	16	2361	2376	2	
frame_centre_horizontal_offset_2	16bit ulmsbf	16	2377	2392	2	

フロントページの続き

(51) Int. Cl. 7

識別記号

F I

テマコード(参考)

H 0 4 N 7/32



F ターム (参考) 5C053 FA03 FA14 FA22 GB06 GB08  
GB17 GB21 GB29 GB37 LA15  
5C059 KK01 KK35 KK39 KK41 MA01  
MA15 MA23 PP05 PP06 PP07  
RC00 RC24 RC31 RE09 SS02  
TA16 TB03 TC27 TC37 TC41  
TC42 UA02 UA05  
5C063 AB03 AC01 CA11 CA23 DA07  
DA13 DB01